

3-5 認識

東京大学 数理・情報教育研究センター
2021年5月1日

概要

- 本節では、AIにおける認識の概念を学び、画像と音声について実社会のどのような場面で活用されているかを知ること、またその技術を実現するためにどのような処理が行われているかを知ることを目指します。
- 物体や音といった実世界の現象をコンピュータに取り込むとどのようなデータになるのか、その性質や注意すべき点についても理解します。

本教材の目次

1. AIにおける認識とは	4
2. 認識技術の活用事例	1 2
- 画像認識の活用事例	1 2
- 音声認識の活用事	2 4
3. 画像処理・認識技術	2 9
- デジタル画像の表現	2 9
- 二次元デジタルフィルタによる画像処理	3 6
- 深層学習による画像認識	4 8
4. 音響処理・音声認識技術	5 6
- デジタル音の表現	5 6
- 周波数解析	6 4
- 音声認識	
5. 認識技術の応用	7 8
	8 6

1. AIにおける認識

AIにおける認識とは

- ・ 「認識」とは「人間（主観）が事物（客観・対象）を認め、それとして知るはたらき。(ref. weblio辞書「認識」)」
- ・ AIが行う認識は「パターン」を認識すること：パターン認識
- ・ パターンとは名前が与えられるような規則的な存在
例：「りんご」はだいたい赤くて丸いという性質、「〇〇さんの顔」と見分けられる画像特徴、「あ」と発話した音声を聞き分ける規則、特定銘柄の株価が見せる特有の変動、ブランドのロゴ、QRコード
- ・ 情報学におけるパターン認識とは、人間が共通概念に基づき名前をつけたサンプルあるいはサンプル集合を模範として、未知のサンプルに対し機械が自動的に名前を与えること
 - ・ 同じ名前を与えられるサンプルの集合は同じクラスに属する
 - ・ クラスにはラベルと呼ばれる名前が与えられている（例：「りんご」）
 - ・ AIにおける認識器は、未知のサンプルを入力すると、それを適切なクラスに分類して、そのラベルを出力する

cf.) 「識別」は対象が同一かを見分けること、「検出」は対象があればそれを見つけること、「分類」は対象を正しいクラスに振り分けること

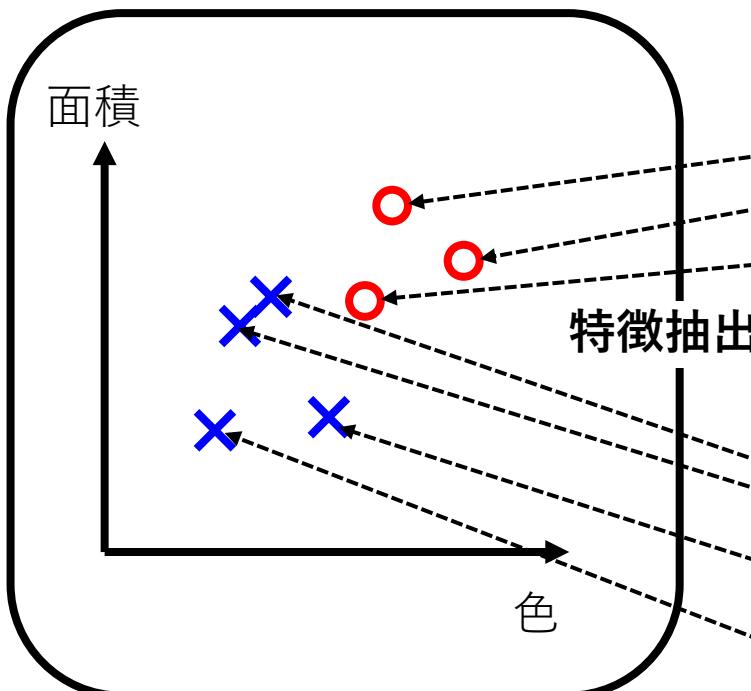
画像認識とは

- 画像に写りこんでいる物体や動作、風景等に対し、人間が共通して与えるであろう名前（ラベル）を特定すること
 - 物体の名前を特定する（リンゴが写っている画像を入力すると「リンゴ」というラベルを返す）：一般物体認識
 - 人間の動作を特定する（人が手を振っている写真や動画を入力すると「手を振る」というラベルを返す）：動作認識
 - 写真の中から「顔」の領域を見つけ出す：顔検出
- 機械学習に基づく「画像認識」とは
 - あらかじめ、認識対象とする各クラスに対し、それと認識されるべき画像の集合が与えられている（訓練データ； training data）
 - データが与えられていないクラスの画像認識は基本的にはできない（ただし、未知のクラスをunknownとして認識する場合もある）
 - どのクラスかわからない画像を適切なクラスに分類する問題なので、「画像分類（Image classification）」と表現されることもある
 - 訓練データに含まれない、クラスが未知の画像集合（評価データ； test data）をどれくらい正しく分類できたかで精度を評価（詳しくは「教師あり学習」を参照）

パターン認識

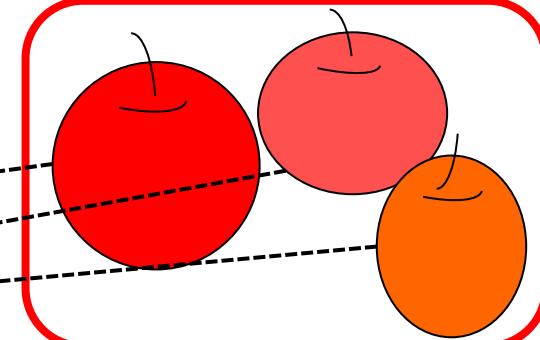
ラベルが既知のサンプル

特徴空間

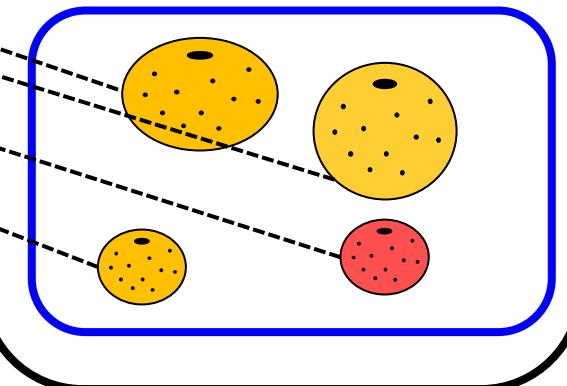


※特徴空間の軸にはこれ以外にも
いろいろなものが考えられる

「りんご」のクラス



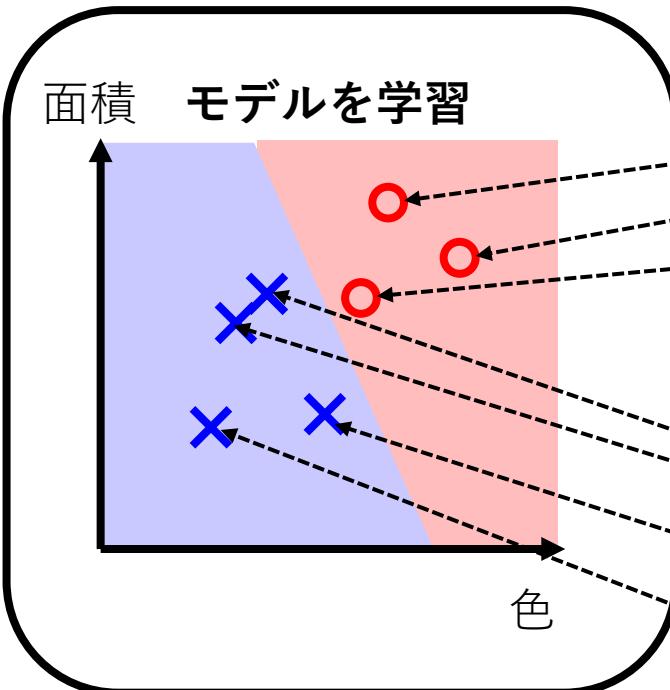
「みかん」のクラス



パターン認識

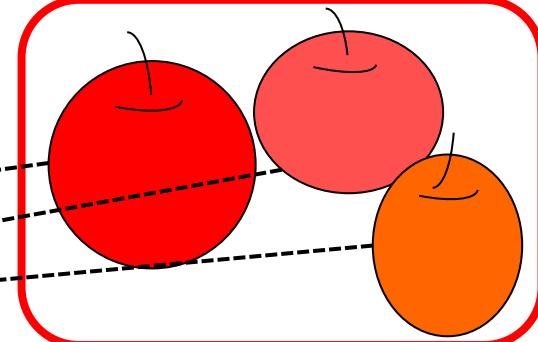
ラベルが既知のサンプル

特徴空間

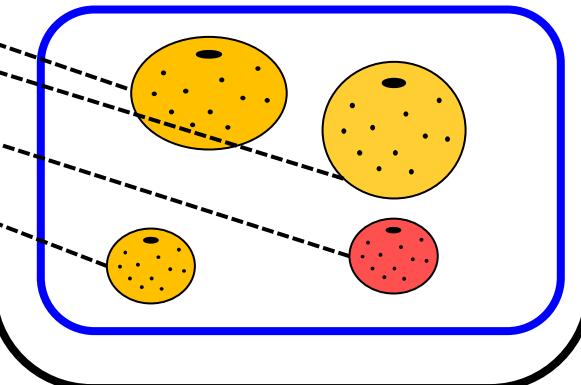


※特徴空間の軸にはこれ以外にもいろいろなものが考えられる

「りんご」のクラス



「みかん」のクラス

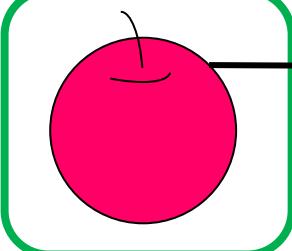


パターン認識

ラベルが既知のサンプル

特徴空間
ラベルが未知の
サンプル

入力

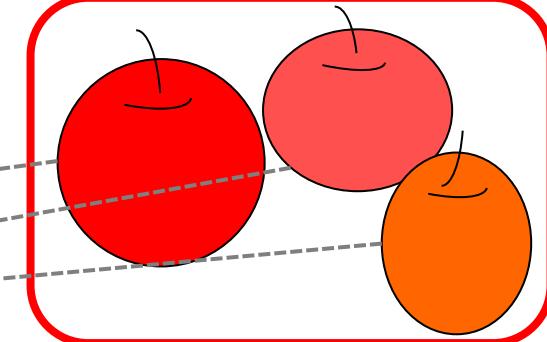


面積 モデル
↑
特徴抽出

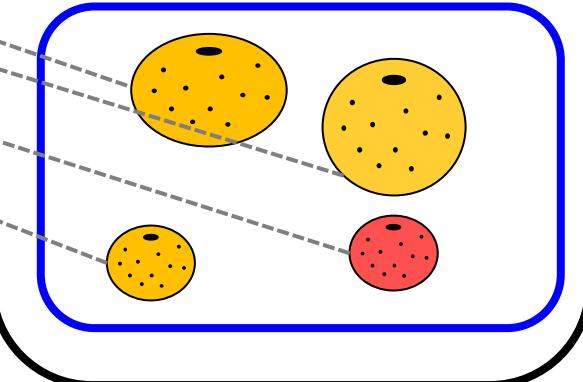
※特徴空間の軸にはこれ以外にも
いろいろなものが考えられる

特徴空間

「りんご」のクラス



「みかん」のクラス

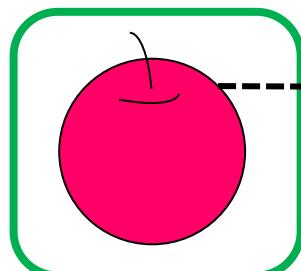


パターン認識

ラベルが既知のサンプル

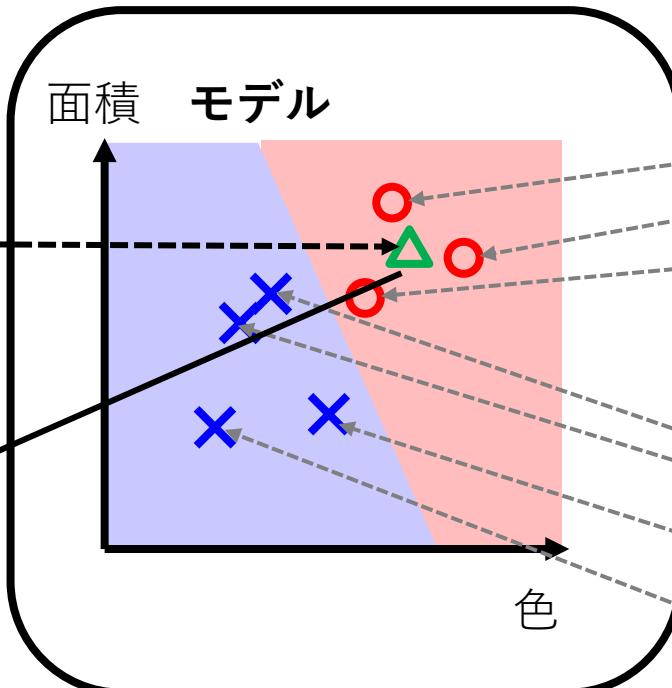
特徴空間

ラベルが未知の
サンプル

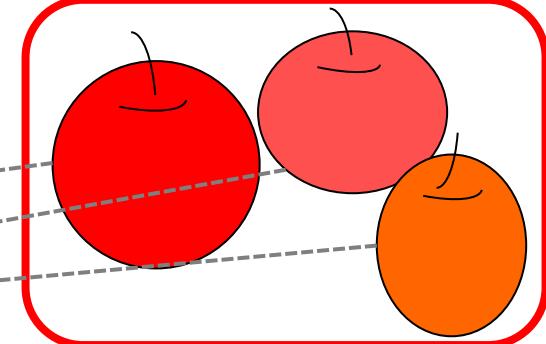


入力

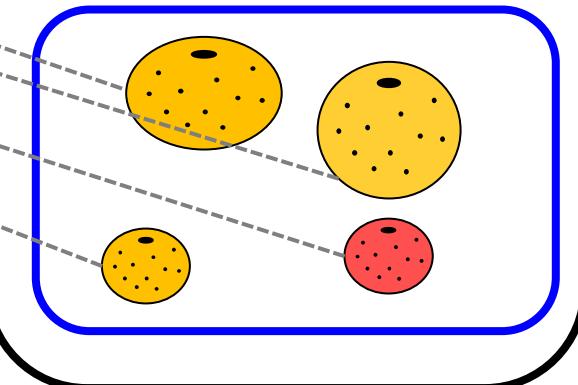
「りんご」
認識結果
(ラベル)



「りんご」のクラス



「みかん」のクラス

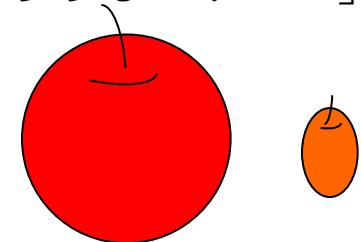


パターン認識の難しさ

目的：ラベルが未知のサンプルを正しいクラスに分類したい

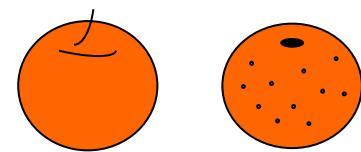
- 難しさ1：クラス内分散が大きい
 - 同じクラスに属するサンプルなのに、互いの特徴が大きく異なる
 - 同じクラスのサンプルが特徴空間中で広い範囲に散らばる

見た目は全然違うのにどちらも「りんご」



- 難しさ2：クラス間分散が小さい
 - 異なるクラスに属するサンプルなのに特徴が似通っている
 - 異なるクラスのサンプルが特徴空間中で互いに近接して分布する

見た目はそっくりなのに「りんご」と「みかん」



- 異なるクラスに属するサンプルの分布が互いに交わると、その交わった領域に位置するサンプルが入力された場合に分類が定まらない
- クラス内分散が小さく、クラス間分散が大きくなるような特徴空間を見つけることが重要

2. 認識技術の活用事例

2.1 画像認識の活用事例

顔領域検出

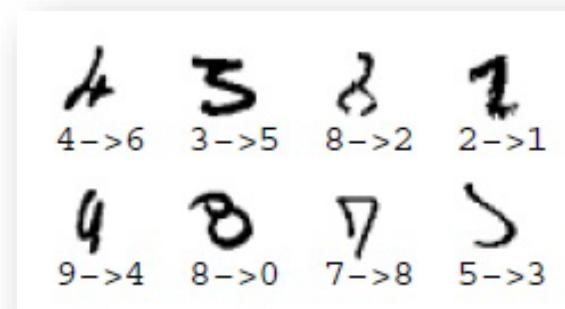
- 写真の中で「顔」らしい領域を検出する
- カメラが自動的に顔にフォーカスを合わせる際にも用いられる



ref. (2020/4/6): Wikimedia commons: File:Kasahara Saitama Kasahara Jinjo Elementary School 1920 1.jpg パブリックドメイン
https://ja.wikipedia.org/wiki/%E3%83%95%E3%82%A1%E3%82%A4%E3%83%AB:Kasahara_Saitama_Kasahara_Jinjo_Elementary_School_1920_1.jpg

手書き文字認識

- 画像認識初期のタスク
- よく用いられるデータセット：MNIST (Mixed National Institute of Standards and Technology database)
 - 「0~9」の10種類の数字の認識
= 10クラス分類タスク
 - 各画像に数字が1つ記入
 - 解像度は28x28 pixel
 - 訓練データ：60,000枚
 - 評価データ：10,000枚
- 間違いややすいサンプルを含む



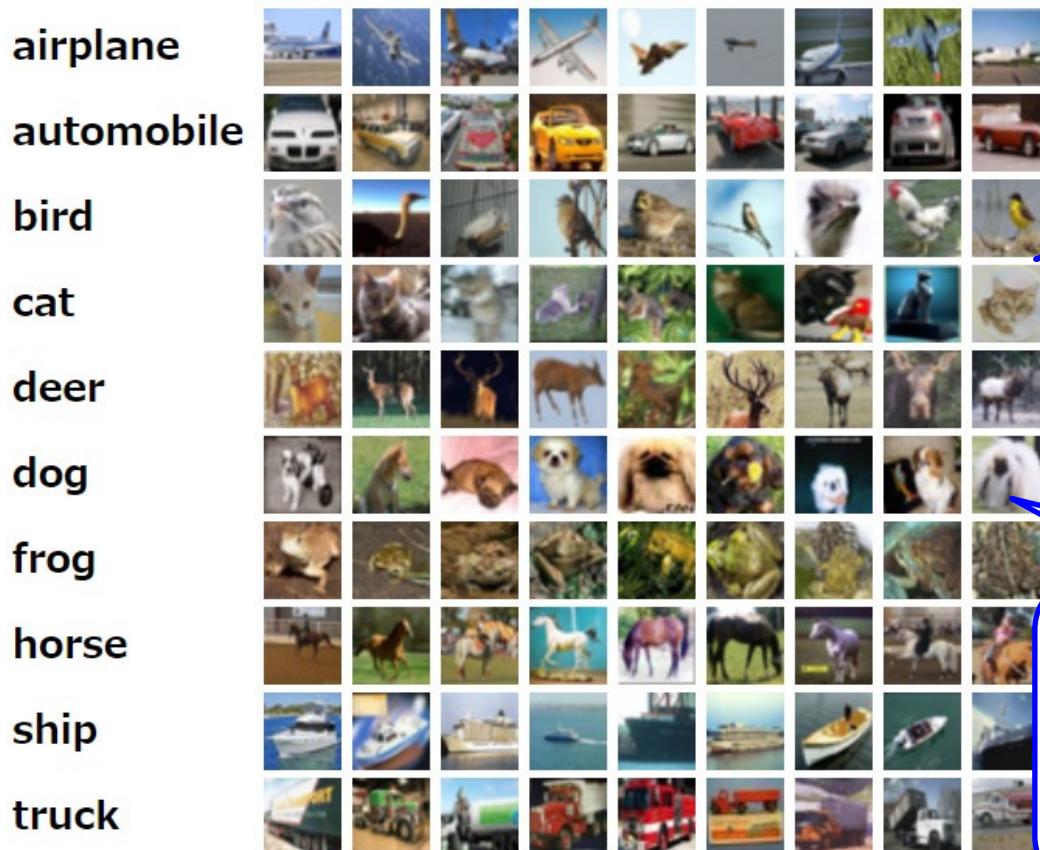
3	6	8	1	7	9	6	6	9	1
6	7	5	7	8	6	3	4	8	5
2	1	7	9	7	1	2	8	4	5
4	8	1	9	0	1	8	8	9	4
7	6	1	8	6	4	1	5	6	0
7	5	9	2	6	5	8	1	9	7
2	2	2	2	2	3	4	4	8	0
0	2	3	8	0	7	3	8	5	7
0	1	4	6	4	6	0	2	4	3
7	1	2	8	7	6	9	8	6	1

ref. (2020/4/6): THE MNIST DATABASE of handwritten digits <http://yann.lecun.com/exdb/mnist/>

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.

一般物体認識

- 写真に写っている物体が何かを当てるタスク
- よく使われるデータセット：CIFAR-10
 - 10クラス分類、各クラス6000枚、32x32 pixelのカラー画像



1枚1枚の画像サイズが小さく
認識対象のクラス数も10種類と
かなり小さなデータセット

基本的には各写真には認識対象の
クラスの物体が1種類だけしか登
場しない
→ ラベルを1つ当てればいいので
比較的簡単なタスク

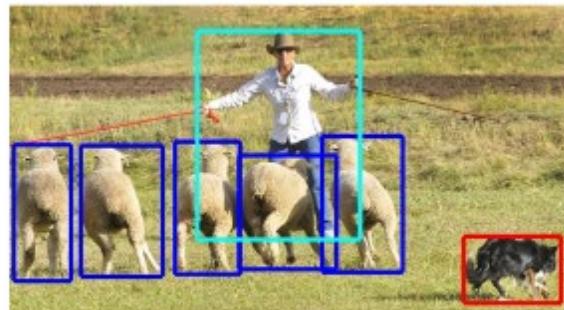
ref. (2020/04/06): The CIFAR-10 dataset, <http://www.cs.toronto.edu/~kriz/cifar.html>

様々な画像認識タスク

- 各画像に対しクラスを1つ特定する物体認識タスク以外にも、様々な画像認識タスクがある
- 多くの研究グループがデータセットを提供（下図はMicrosoft COCO）

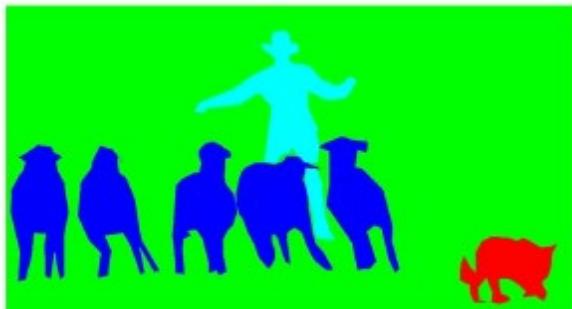


(a) Image classification

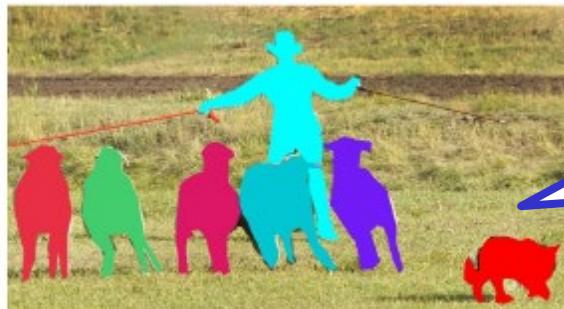


(b) Object localization

- (a) 複数物体認識
- (b) 矩形領域検出
- (c) 領域セグメンテーション
- (d) 個体別領域セグメンテーション



(c) Semantic segmentation



(d) This work

個体ごとに
別の領域として抽出

画像の動作認識 (Action recognition)

- 物体ではなく動作を認識するタスク
- 下図はPASCAL VOCのAction Classification Competitionの例
 - 10クラス(ジャンプする、電話する、楽器を演奏する、読む、自転車やバイクに乗る、馬に乗る、走る、写真を撮る、パソコンを使う、歩く)
 - 上の10クラスに属さない動作を行う画像 ("その他")も含まれている

10 action classes + "other"



ref. (2020/4/6) The PASCAL Visual Object Classes Challenge 2012 (VOC2012) <http://host.robots.ox.ac.uk/pascal/VOC/>

東京大学 山肩洋子 2020 CC BY-NC-SA

動画における動作認識 (Action recognition)

- 画像ではなく動画を対象とした動作認識
- ショートクリップに写っている人物の動作を認識
- 下図はUCF101: University of Central Floridaの研究グループが作った、101種類の動作認識を行うタスクのためのデータセットの例
 - 合計13320クリップ
 - 各動画の平均長は7.21 s
(最短1.06 s、最長71.04 sec)
 - 解像度320x240
 - うち51種類については音付



UCF101: A Dataset of 101 Human Action Classes From Videos in The Wild

https://www.crcv.ucf.edu/wp-content/uploads/2019/03/UCF101_CRCV-TR-12-01.pdf

画像に対するキャプション生成 (Image captioning)

- 画像に対し、その内容を説明する短い文 (caption)を生成するタスク
- データセットの一例：MS-COCO
 - 123,287枚の画像に886,284個の物体領域
 - 個々の物体ごとの領域とその物体ラベル
 - 5文程度のキャプション



ref. (2020/4/6) Microsoft COCO, <http://cocodataset.org/>

物体ラベル：

“person”, “car”, “dog”, “sheep”, “chair”

キャプション：

- the dog is hurdling the sheep towards the man.
(その犬は羊を男のほうに追いやっている)
- a man and a dog herding some sheep in a fenced off area.
(男と犬が柵で囲まれた場所で羊を群れにしている)
- a dog chases after some goats while a man watches.
(男が見守る中、犬がヤギを追いかける)
- a man guides a dog to herd sheep.
(男が犬を誘導して羊の群れを作る)
- a dog herds sheep into a pen as a shepherd looks on
(犬は羊飼いが見ている間に、羊を小さな群れにする)

画像に対するキャプション生成 (Image captioning)

深層学習により自動生成されたキャプションの例

(ありがちな情景と誤認して生成された誤ったキャプションがあることに注意)



黒い服を着た男性が
ギターを弾いている



"construction worker in orange
safety vest is working on road."



二人の小さな女の子が
レゴで遊んでいる



"boy is doing backflip on
wakeboard."



"girl in pink dress is jumping in
air."



白と黒の犬が
棒の上を跳んでいる



ピンクのシャツを着た
女の子がブランコで揺れている



blue wetsuit is surfing on
wave."

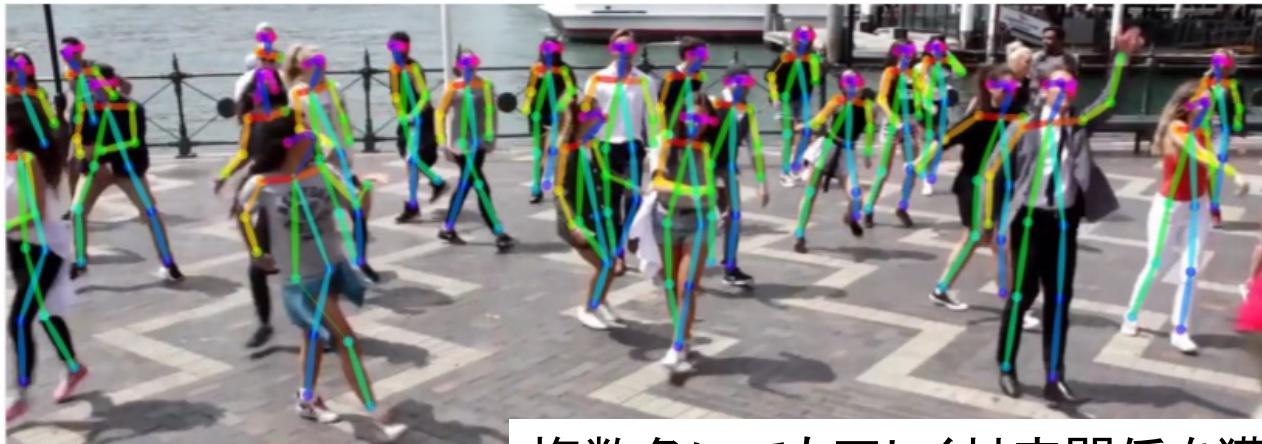
ref. (2020.4.6) Andrej Karpathy, Li Fei-Fei, "Deep Visual-Semantic Alignments for Generating Image Descriptions", CVPR2015.

<https://cs.stanford.edu/people/karpathy/deepimagesent/>

人体の姿勢推定

OpenPose: 米国カーネギーメロン大学が開発

- 人体の19か所の関節（左右の区別あり）を高精度で検出
- 商用にも広く使われている



複数名いても正しく対応関係を獲得



右の肘と右の手首の連結の尤度マップ 位置と方向を検出

ref. (2020/4/6) Zhe Cao, Tomas Simon, Shih-En Wei, Yaser Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields", CVPR2017,
<https://github.com/CMU-Perceptual-Computing-Lab/openpose>

顔認証

- 顔画像から特定の個人であるかどうかを判別
 - PCやモバイル機器のロック解除のための認証に利用
- 通常のカメラで撮影したカラー画像に加え赤外線画像を利用(注1)
 - カラー画像は光源の位置や明るさが変わると、同一人物であっても画像特徴が大きく変化する
 - 赤外線カメラは照明の違いによる影響を受けづらい
→ 顔認証ではカラー画像と赤外線画像を両方使って顔認証を行う場合が多い
- 深層学習により顔画像認証 (Deep face recognition)
 - たくさんの人の顔でモデルを学習
 - そうして得られたモデルで顔から特徴を抽出して同一人物かを評価
 - 空港の監視カメラで撮影した画像など、顔がカメラを向いているとは限らない場合は、顔の向きが違っても同一人物かを見分けることが重要

注1) (2021/3/31): "Windows Hello 顔認証". Windows ハードウェア開発者向けドキュメント, <https://docs.microsoft.com/ja-jp/windows-hardware/design/device-experiences/windows-hello-face-authentication>

指紋認証

- ・ 指紋をスキャンして獲得した画像から、指紋の特徴を抽出し、登録済みのパターンと照らし合わせることにより生体認証
- ・ iPhone/iPadにもTouch IDと呼ばれる指紋認証が内蔵
 - ・ 静電容量式のタッチセンサーが指紋の細かい部分の表皮下の層から高解像度画像を取得
 - ・ 指紋を基本的な3つの種類(弓状、蹄状、渦状)のいずれかに分類
 - ・ それらの特徴の組み合わせにより個人認証



弓状紋



蹄状紋



渦状紋

ref. (2021/3/31): File:Fingerprint Arch.jpg, パブリックドメイン, https://commons.wikimedia.org/wiki/File:Fingerprint_Arch.jpg

ref. (2021/3/31): File:Fingerprint Loop.jpg, パブリックドメイン, https://commons.wikimedia.org/wiki/File:Fingerprint_Loop.jpg

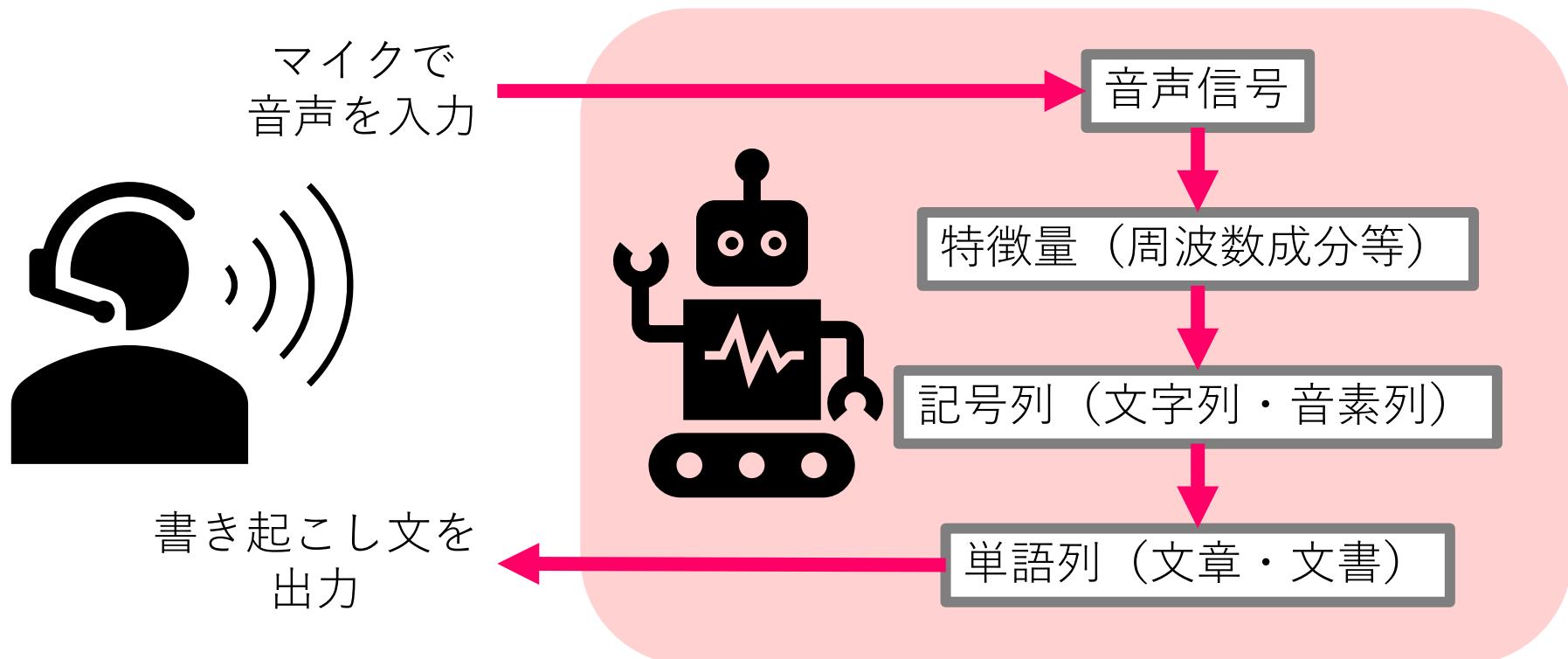
ref. (2021/3/31): File:Fingerprint Whorl.jpg, パブリックドメイン, https://web.archive.org/web/20050403155444/http://www.nist.gov/srd/fing_img.htm

2. 認識技術の活用事例

2.1 音声認識の活用事例

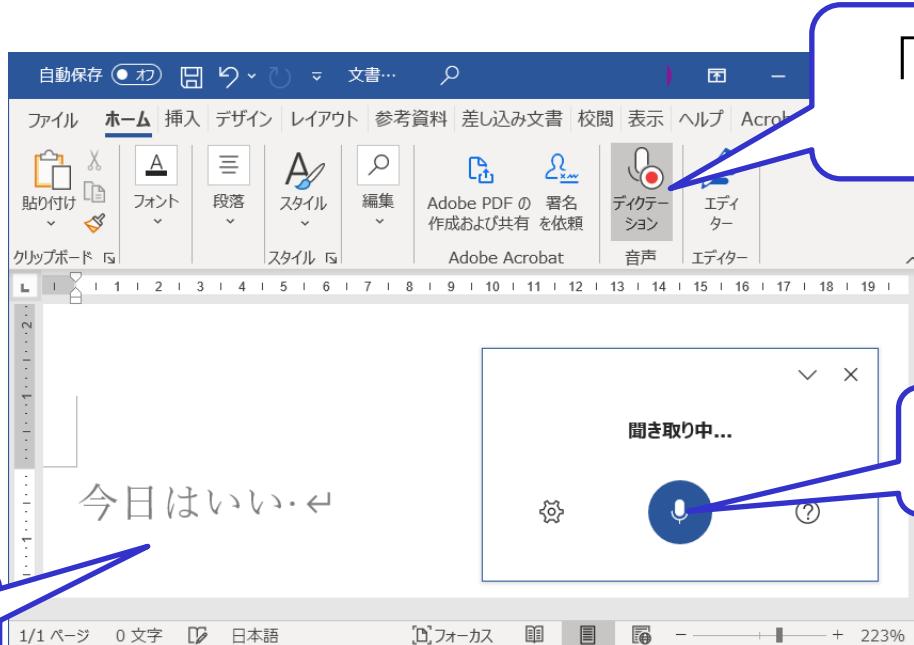
音声認識における処理の流れ

Speech-to-Text (STT) とも呼ぶ (cf. TTS; Text-to-Speech、音声合成)



音声書き起こし・ディクテーション

- WordやGoogleドキュメントなどにも内蔵
- 会議や講演、講義の記録作成
 - 衆議院の本会議・委員会審議の議事録作成（2011～）
 - 聴覚障害者のための情報保証にも使われる



リアルタイムで
書き起こし

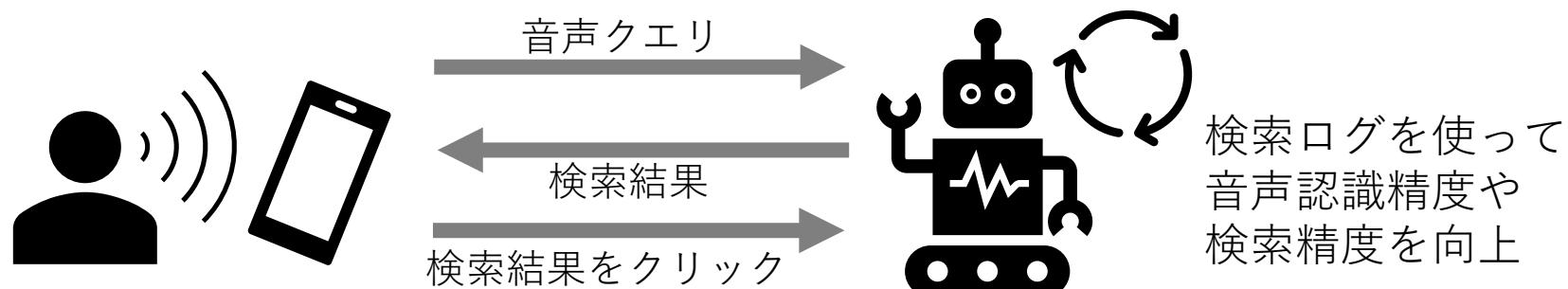
動画の自動字幕生成

- 動画の音声を認識して字幕を自動付与する機能
 - youtubeにも内蔵
(2009年に英語でスタート、2011年から日本語に対応)
 - 機械翻訳との組み合わせも可能



音声検索・スマートスピーカ・音声対話システム

- 2018年、世界のオンライン人口の27%が音声検索を利用 [1]
- 2019年、米国ではインターネット利用者の26.0%（約7千4百万人）がスマートスピーカを利用 [2]
- 音声検索では、ユーザが音声で入力したクエリとユーザの検索結果に対する選択行動の記録（検索ログ）を収集し、認識モデルを改良
→ 使えるほど音声データが集まり音声認識精度が向上する仕組み
- スマートスピーカでは、高度な発話意図理解と対話が求められる
 - 一問一答型：「今日の天気は？」 → 場所が「ここ」であることは暗黙の了解
 - 続けて対話をするために対話管理が必要：
ユーザ「明日の東京の天気は？」 → AI「晴れです。」
→ ユーザ「最低気温は？」 → AI（「明日の東京の」と解釈し）「3度です。」



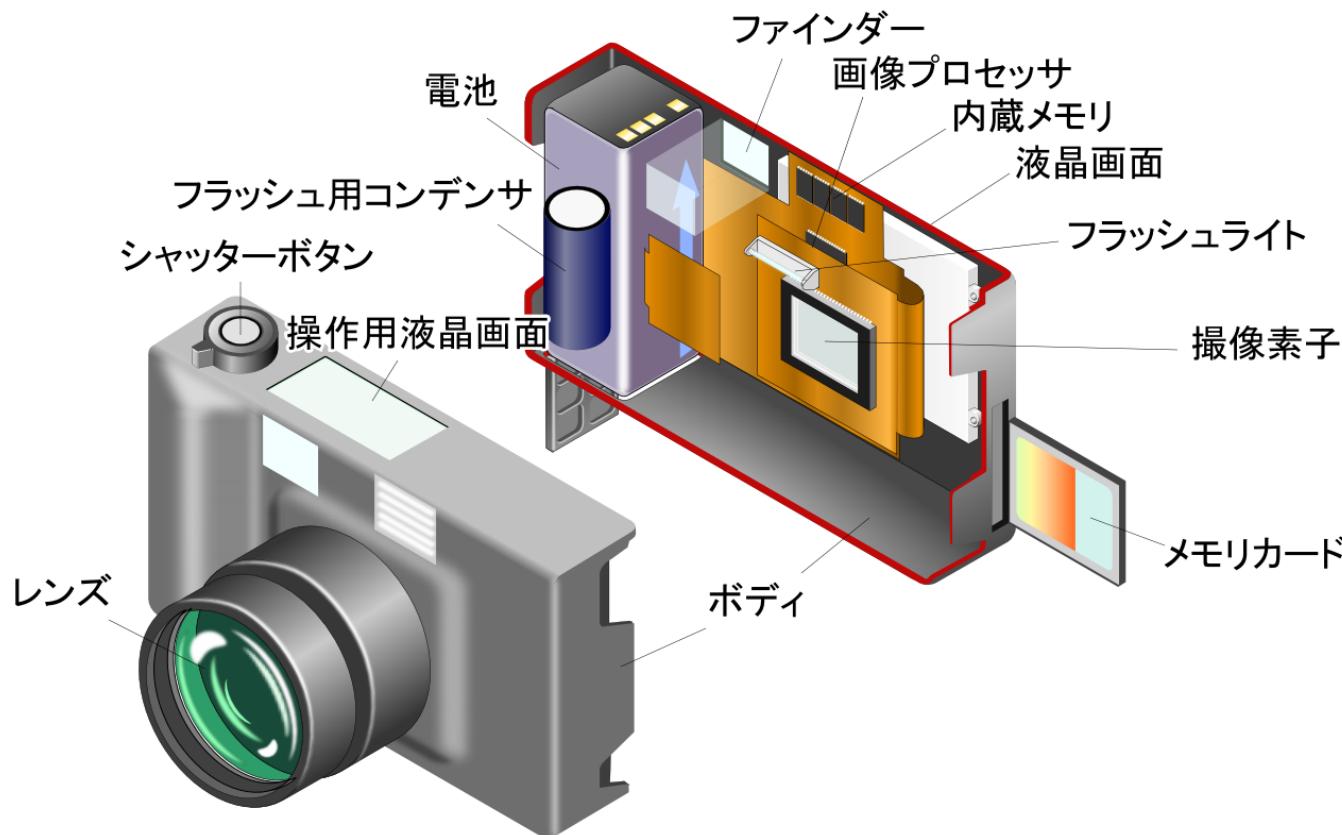
[1] <https://www.thinkwithgoogle.com/marketing-strategies/search/voice-search-mobile-use-statistics/>

[2] ref. Corey McNair, "Global Smart Speaker Users 2019—Trends for Canada, China, France, Germany, the UK and the US —", INSIDER INTELLIGENCE, <https://www.emarketer.com/content/global-smart-speaker-users-2019>

3.画像処理・認識技術

3.1 デジタル画像の表現

デジタルカメラの仕組み



ref:千葉憲明著、『カメラの常識のウソ・マコト』、講談社、2004年6月20日第1刷発行、ISBN 4062574462, 森枝卓士著、『デジカメ時代の写真術』、NHK出版、2003年7月10日第1刷発行、ISBN 4140880740, 津軽海渡、木村誠恥著、『図解雑学 デジタルカメラ』、ナツメ社、2002年12月18日発行、ISBN 4816334092

ref. (2020/4/3): Wikimedia commons: File:Digital camera cut model 1.PNG (https://commons.wikimedia.org/wiki/File:Digital_camera_cut_model_1.PNG) CC BY-SA 3.0

アナログ画像とデジタル画像の表現

画像とは x 軸と y 軸からなる平面座標系で定義される2変数関数 f または F

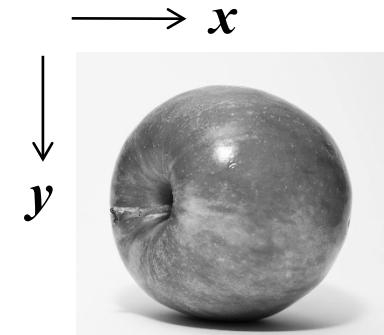
アナログ画像の表現

$$z = f(x, y)$$

x, y, z は0以上の実数

(x, y) は平面上の位置座標

z は (x, y) における光の強さ・明るさ



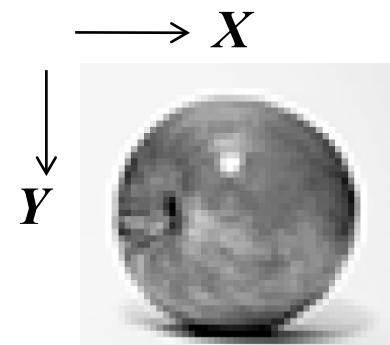
デジタル画像の表現

$$Z = F(X, Y)$$

X, Y, Z は0以上の整数

(X, Y) は平面上の格子点

Z は格子点 (X, Y) における光の強さ・明るさ

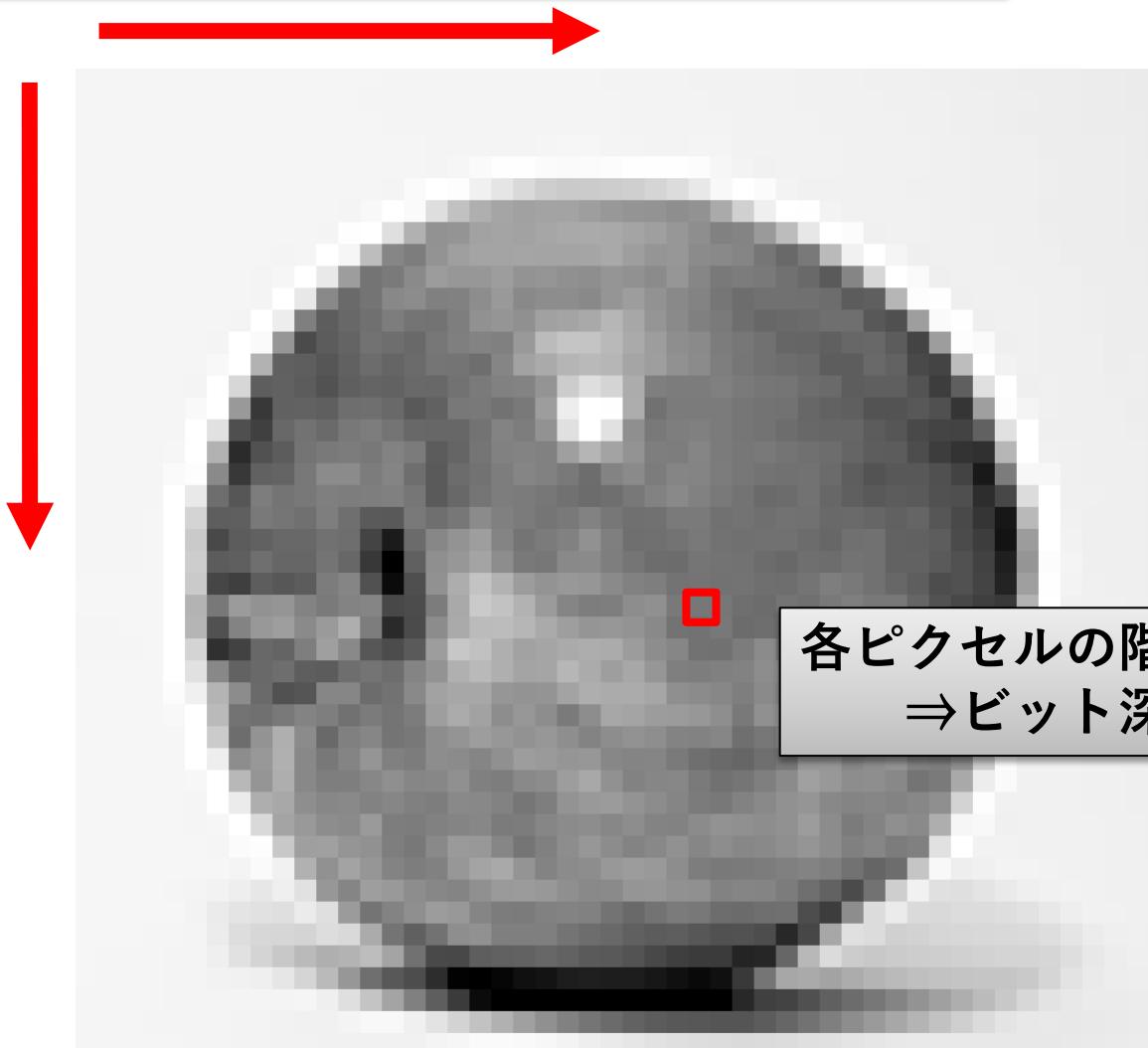


1つの格子点を画素またはピクセル(pixel)と呼ぶ
色の濃さ(Z の値の大きさ)を輝度または濃淡値と呼ぶ

ref. (2020/4/3): Wikimedia commons:
File:Red Apple.jpg CC BY 2.0
[https://commons.wikimedia.org/wiki/
File:Red_Apple.jpg](https://commons.wikimedia.org/wiki/File:Red_Apple.jpg)

デジタル画像は正方形タイルの集まり

横縦方向の刻みが細かい \Rightarrow 解像度が高い

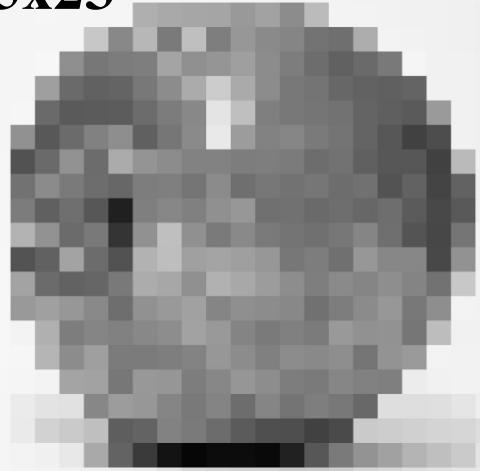


ref. (2020/4/3): Wikimedia commons:
File:Red Apple.jpg CC BY 2.0
https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

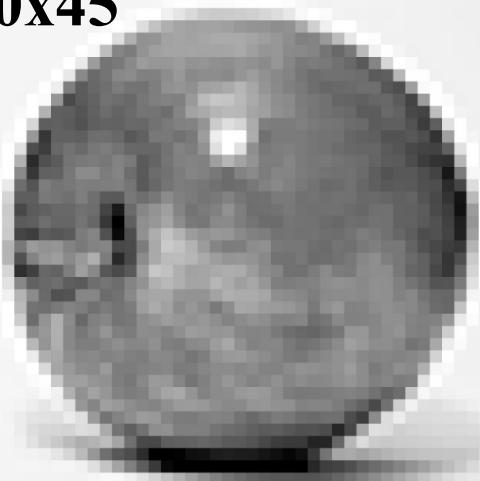
解像度の高低による画像の違い

ref. (2020/4/3): Wikimedia commons:
File:Red Apple.jpg CC BY 2.0
https://commons.wikimedia.org/wiki/File:Red_Apple.jpg

25x23



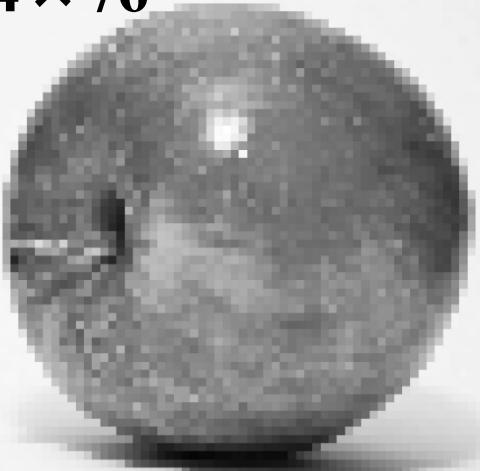
50x45



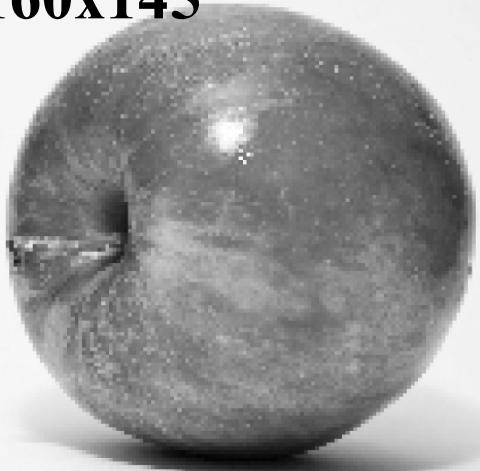
低解像度



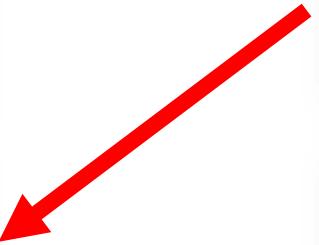
84 × 76



160x145



高解像度



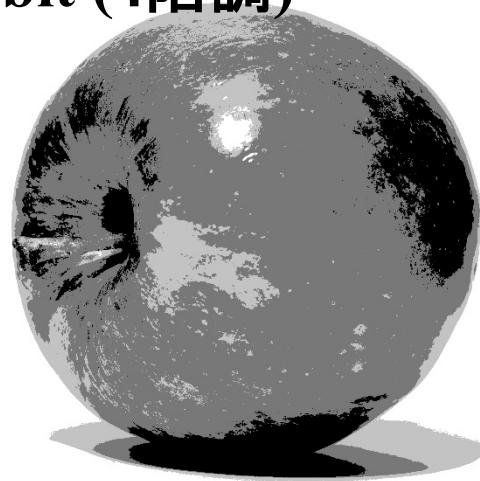
ビット深度の深さによる画像の違い

1bit (2階調):

二値画像 (binary image)
とも呼ぶ



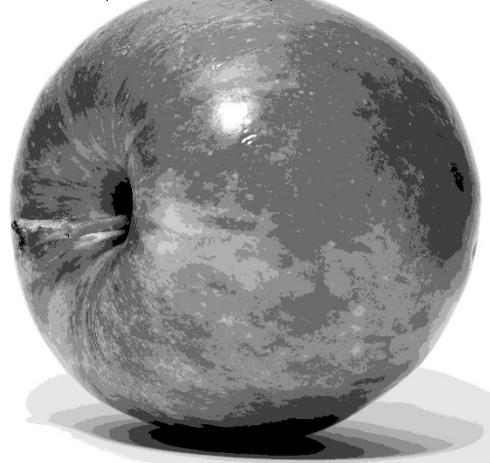
2bit (4階調)



低ビット深度



3bit (8階調)



高ビット深度

8bit (256階調):



一般的にグレースケール
画像といえばこれ

デジタル画像（24bitフルカラー）の表現

R 1画素あたり8bit
(256階調)



+

G 1画素あたり8bit
(256階調)



+

B 1画素あたり8bit
(256階調)



カラー画像はRGBの
3枚の濃淡画像として
記録されている

=



1画素あたり24bit
(1677万色)
フルカラー

3.画像処理・認識技術

3.2 2次元フィルタリングによる 画像処理

置み込み演算による画像のフィルタリング

- デジタル画像を平面座標系において座標 (x, y) が決まると、その座標における濃淡値 $z=f(x,y)$ を返す関数と考える（ただし、 x, y, z は0以上の整数）
- 入力画像 $f(x,y)$ に対し、何らかのフィルタを適用して、出力画像 $g(x,y)$ を生成することを考える

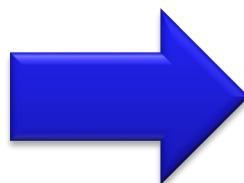
入力画像 $f(x,y)$



出力画像 $g(x,y)$



ケーススタディ:
画像を
ぼけさせる
フィルタとは？



2次元デジタルフィルタにおける畳み込み演算

$h(0,0)$	$h(0,1)$	$h(0,2)$
$h(1,0)$	$h(1,1)$	$h(1,2)$
$h(2,0)$	$h(2,1)$	$h(2,2)$

3×3のカーネル

$$g(n, m) = \sum_i \sum_j h(i, j) f(n + i, m + j)$$

出力画像の明るさを変えたくない場合は、

$$\sum_i \sum_j h(i, j) = 1$$

- 一般的にフィルタのカーネルは奇数×奇数画素の正方行列
- それぞれの画素を中心とし、その画素とその周辺からなる画素集合とカーネルとの行列積を、新しい画像の画素値とする処理を「畳み込み (convolution)」と呼ぶ
- 畳み込みはDeep Learningによる画像処理の基本的な処理 (CNN=Convolutional Neural Network)
- フィルタによって出力画像が様々に変化

2次元デジタルフィルタの適用

入力画像 $f(x,y)$

$f(0,0)$	$f(0,1)$	$f(0,2)$	$f(0,3)$	$f(0,4)$	$f(0,5)$		
$f(1,0)$	$f(1,1)$	$f(1,2)$	$f(1,3)$	$f(1,4)$	$f(1,5)$		
$f(2,0)$	$f(2,1)$	$f(2,2)$	$f(2,3)$	$f(2,4)$	$f(2,5)$		
$f(3,0)$	$f(3,1)$	$f(3,2)$	$f(3,3)$	$h(0,0)$	$h(0,1)$	$h(0,2)$	
$f(4,0)$	$f(4,1)$	$f(4,2)$	$f(4,3)$	$h(1,0)$	$h(1,1)$	$h(1,2)$	
$f(5,0)$	$f(5,1)$	$f(5,2)$	$f(5,3)$	$h(2,0)$	$h(2,1)$	$h(2,2)$	

出力画像 $g(x,y)$

行列の積

$$g(n, m) = \sum_{i=-1}^1 \sum_{j=-1}^1 h(i, j) f(n + i, m + j)$$

2次元デジタルフィルタの適用

入力画像 $f(x,y)$

$f(0,0)$	$f(0,1)$	$f(0,2)$	$f(0,3)$	$f(0,4)$	$f(0,5)$	$g(0,0)$	$g(0,1)$	$g(0,2)$	$g(0,3)$	$g(0,4)$
$f(1,0)$	$f(1,1)$	$f(1,2)$	$f(1,3)$	$f(1,4)$	$f(1,5)$	$g(1,0)$	$g(1,1)$	$g(1,2)$	$g(1,3)$	$g(1,4)$
$f(2,0)$	$f(2,1)$	$f(2,2)$	$f(2,3)$	$f(2,4)$	$f(2,5)$	$g(2,0)$	$g(2,1)$	$g(2,2)$	$g(2,3)$	$g(2,4)$
$f(3,0)$	$f(3,1)$	$f(3,2)$	$f(3,3)$	$f(3,4)$	$h(0,0)$	$h(0,1)$	$h(0,2)$	$g(3,0)$	$g(3,1)$	$g(3,2)$
$f(4,0)$	$f(4,1)$	$f(4,2)$	$f(4,3)$	$f(4,4)$	$h(1,0)$	$h(1,1)$	$h(1,2)$	$g(4,0)$	$g(4,1)$	$g(4,2)$
$f(5,0)$	$f(5,1)$	$f(5,2)$	$f(5,3)$	$f(5,4)$	$h(2,0)$	$h(2,1)$	$h(2,2)$	$g(5,0)$	$g(5,1)$	$g(5,2)$

$$g(n, m) = \sum_{i=-1}^1 \sum_{j=-1}^1 h(i, j) f(n + i, m + j)$$

2次元デジタルフィルタの適用

入力画像 $f(x,y)$

$f(0,0)$	$f(0,1)$	$f(0,2)$	$f(0,3)$	$f(0,4)$	$f(0,5)$	出力画像 $g(x,y)$				
$f(1,0)$	$f(1,1)$	$f(1,2)$	$f(1,3)$	$f(1,4)$	$f(1,5)$	$g(0,0)$	$g(0,1)$	$g(0,2)$	$g(0,3)$	$g(0,4)$
$f(2,0)$	$f(2,1)$	$f(2,2)$	$f(2,3)$	$f(2,4)$	$f(2,5)$	$g(1,0)$	$g(1,1)$	$g(1,2)$	$g(1,3)$	$g(1,4)$
$f(3,0)$	$f(3,1)$	$f(3,2)$	$f(3,3)$	$f(3,4)$	$f(3,5)$	$g(2,0)$	$g(2,1)$	$g(2,2)$	$g(2,3)$	$g(2,4)$
$f(4,0)$	$f(4,1)$	$f(4,2)$	$f(4,3)$	$f(4,4)$	$f(4,5)$	$g(3,0)$	$g(3,1)$	$g(3,2)$	$g(3,3)$	$g(3,4)$
$f(5,0)$	$f(5,1)$	$f(5,2)$	$f(5,3)$	$f(5,4)$	$f(5,5)$	$g(4,0)$	$g(4,1)$	$g(4,2)$	$g(4,3)$	$g(4,4)$

$$g(n, m) = \sum_{i=-1}^1 \sum_{j=-1}^1 h(i, j) f(n + i, m + j)$$

2次元デジタルフィルタの適用

入力画像 $f(x,y)$

$f(0,0)$	$f(0,1)$	$f(0,2)$	$f(0,3)$	$f(0,4)$	$f(0,5)$
$f(1,0)$	$f(1,1)$	$f(1,2)$	$f(1,3)$	$f(1,4)$	$f(1,5)$
$f(2,0)$	$f(2,1)$	$f(2,2)$	$f(2,3)$	$f(2,4)$	$f(2,5)$
$f(3,0)$	$f(3,1)$	$f(3,2)$	$f(3,3)$	$f(3,4)$	$f(3,5)$
$f(4,0)$	$f(4,1)$	$f(4,2)$	$f(4,3)$	$f(4,4)$	$f(4,5)$
$f(5,0)$	$f(5,1)$	$f(5,2)$	$f(5,3)$	$f(5,4)$	$f(5,5)$

出力画像 $g(x,y)$

$g(0,0)$	$g(0,1)$	$g(0,2)$	$g(0,3)$	$g(0,4)$
$g(1,0)$	$g(1,1)$	$g(1,2)$	$g(1,3)$	$g(1,4)$
$g(2,0)$	$g(2,1)$	$g(2,2)$	$g(2,3)$	$g(2,4)$
$g(3,0)$	$g(3,1)$	$g(3,2)$	$g(3,3)$	$g(3,4)$
$g(4,0)$	$g(4,1)$	$g(4,2)$	$g(4,3)$	$g(4,4)$

$h(i,j)$ のとる値によって
出力画像が様々に変化

$$g(n, m) = \sum_{i=-1}^1 \sum_{j=-1}^1 h(i, j) f(n + i, m + j)$$

→ h の f に対する畳み込み演算と呼ぶ

平均値フィルタ (Mean filter)

画像をぼけさせる
フィルタ

入力画像(320x480)



3x3 平均値フィルタ

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



5x5 平均値フィルタ

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25



鮮鋭化フィルタ

輪郭を際立たせ、鮮鋭にするフィルタ

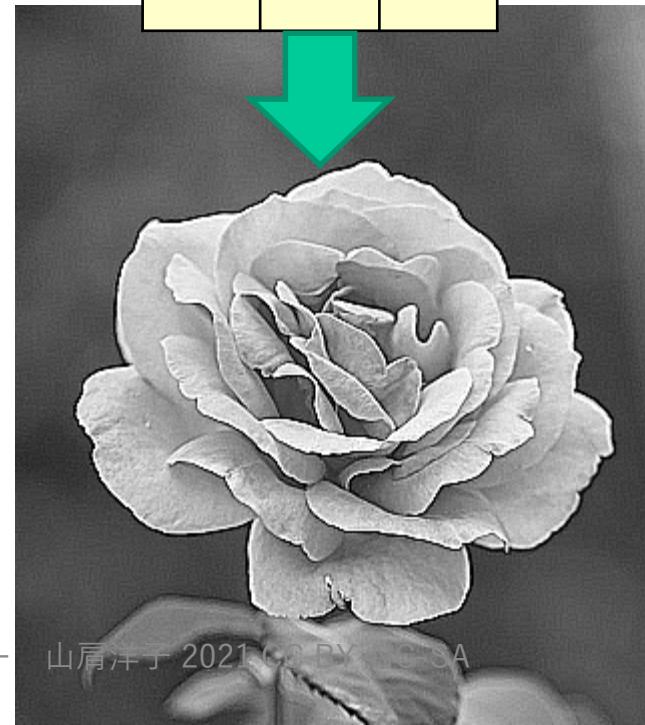
- その画素と周辺の画素の値が違うとき、その画素の値を大きくする
- その画素と周辺の画素の値が同じ時、その画素の元の値をそのまま使う

入力画像 (320x480)



3x3鮮鋭化フィルタ
 $k=5$ とすると
下図のようになる

$-k/8$	$-k/8$	$-k/8$
$-k/8$	$1+k$	$-k/8$
$-k/8$	$-k/8$	$-k/8$



エッジ抽出：ラプラシアンフィルタ

その画素の値が周辺の画素と違うほど大きな値とすることによって、周辺との輝度が大きく変化する輪郭線を取り出す

入力画像(320x480)



3x3 ラプラシアンフィルタ

-1	-1	-1
-1	8	-1
-1	-1	-1



5x5 ラプラシアンフィルタ

-1	-3	-4	-3	-1
-3	0	6	0	-3
-4	6	20	6	-4
-3	0	6	0	-3
-1	-3	-4	-3	-1



ガウシアンフィルタ

中心が最も値が大きく、
中心から離れるほど値が小さくなる
x軸方向、y軸方向それぞれの値の変化は
ガウス関数に従う

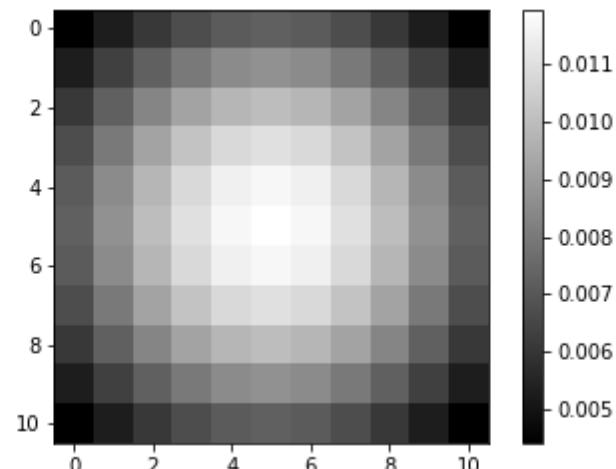
$$h(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right)$$

カメラのレンズによるボケを
幾何学的に再現したモデルと考えられる

サイズ 3×3 、 $\sigma=2$ のフィルタ

0.102	0.115	0.102
0.115	0.131	0.115
0.102	0.115	0.102

11x11のガウシアンフィルタ



サイズ 5×5 、 $\sigma=5$ のフィルタ

0.0369	0.0392	0.0400	0.0392	0.0369
0.0392	0.0416	0.0424	0.0416	0.0392
0.0400	0.0424	0.0433	0.0424	0.0400
0.0392	0.0416	0.0424	0.0416	0.0392
0.0369	0.0392	0.0400	0.0392	0.0369

ガウシアンフィルタ

- その画素の値の影響を最も強く受け、そこから離れた位置にある画素ほど影響を受けなくなる

入力画像(320x480)



ガウシアンフィルタ
サイズ: 5x5
 $\sigma=5$



ガウシアンフィルタ
サイズ: 11x11
 $\sigma=5$



3.画像処理・認識技術

3.3 深層学習による画像認識

深層学習による画像認識の幕開け

- ・ クラウドソーシングによる大規模データセット：ImageNet
- ・ ImageNet Large-Scale Visual Recognition Challenge (ILSVRC)
 - 1000クラス、学習120万枚、検証5万枚、テスト10万枚
 - マルチラベル：1枚の画像に複数ラベル + 信頼度
- ・ 2011年のILSVRCで優勝したモデルのエラー率は26%
- 2012年にDeep Learningを使ったモデルが登場→15%に激減！

Easiest classes

red fox (100) hen-of-the-woods (100) ibex (100) goldfinch (100) flat-coated retriever (100)



tiger (100)

hamster (100)

porcupine (100)

stingray (100)

Blenheim spaniel (100)



Hardest classes

muzzle (71) hatchet (68) water bottle (68) velvet (68)



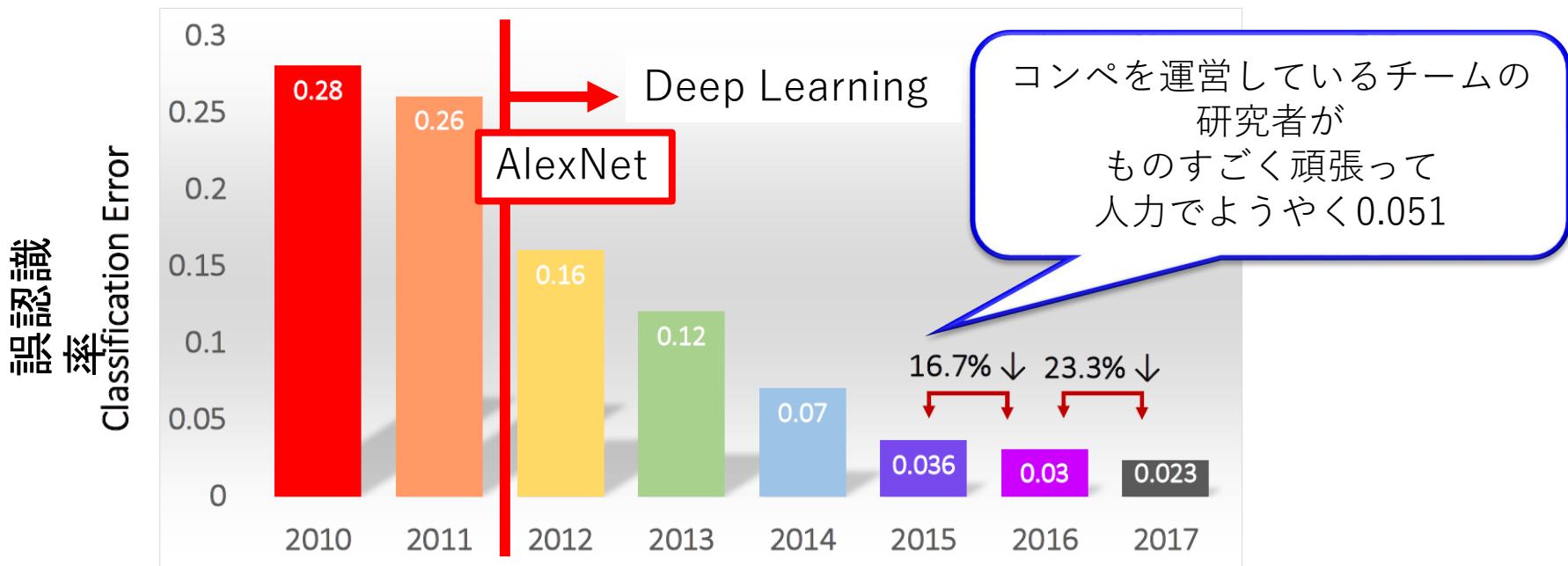
loupe (66)



ref. Russakovsky, O., Deng, J., Su, H. et al. ImageNet Large Scale Visual Recognition Challenge. *Int J Comput Vis* 115, 211–252 (2015).

物体認識精度は人間を超えた!?

- ILSVRC2012で、トロント大学Geoffrey Hinton教授率いるグループが初めて多層ニューラルネットワークによる画像認識モデルを提案
- 主著者の名前をもじってAlexNetと呼ばれる
- 翌年から上位チームはすべてDeep Learningを採用
- 2015年についに人間の精度を超えた

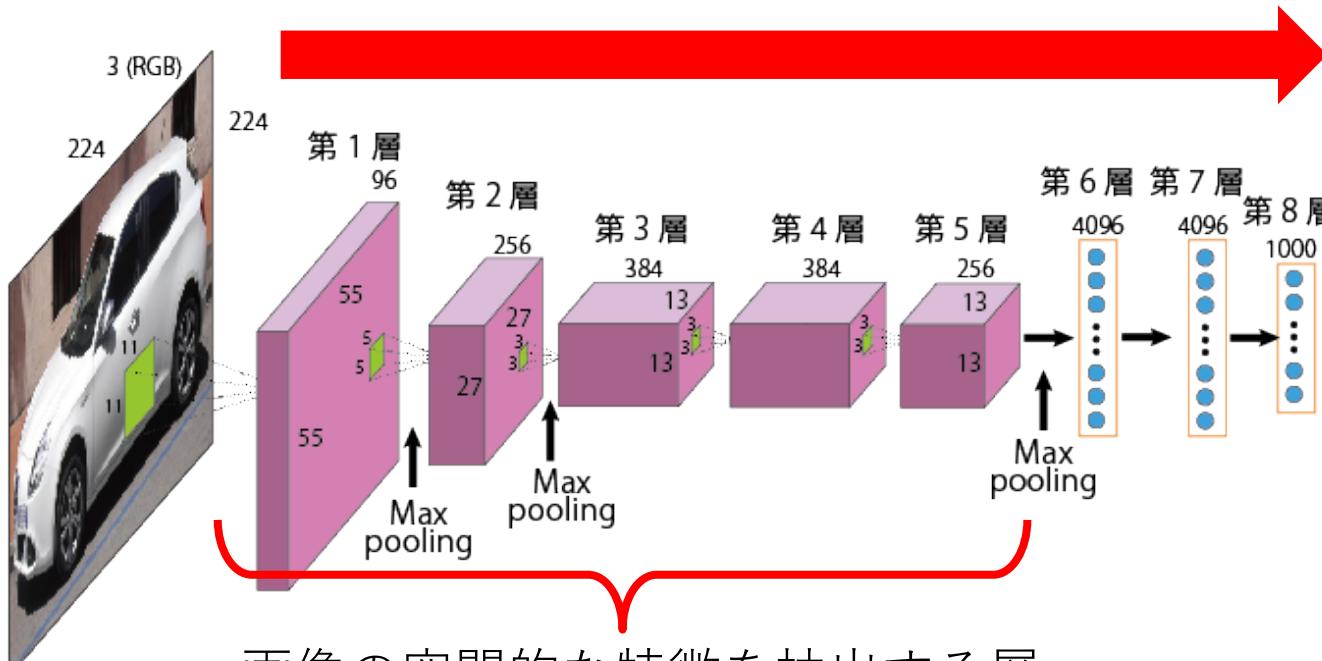


Excerpted from ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2017 Overview

Andrej Karpathy, "What I learned from competing against a ConvNet on ImageNet", Sep 2, 2014, <http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/>

畳み込みニューラルネットワーク (CNN; Convolutional Neural Network)

- 代表的な物体識別モデル
- 実際には様々な実装がある（下図はAlexNet）



- 画像の空間的な特徴を抽出する層
- 途中の層ではどうなっているか？

最終層にsoftmax
を適用した結果

0.00 goldfish
0.00 great white shark
0.00 tiger shark
0.00 hammerhead
0.01 electric ray
...
0.89 car
...
0.00 stinkhorn
0.00 earthstar
0.00 hen-of-the-woods
0.02 bolete
0.00 ear, spike
0.00 toilet tissue

1000クラスのうちcarだけが高く、
残りがほぼ0のようなベクトルが出力

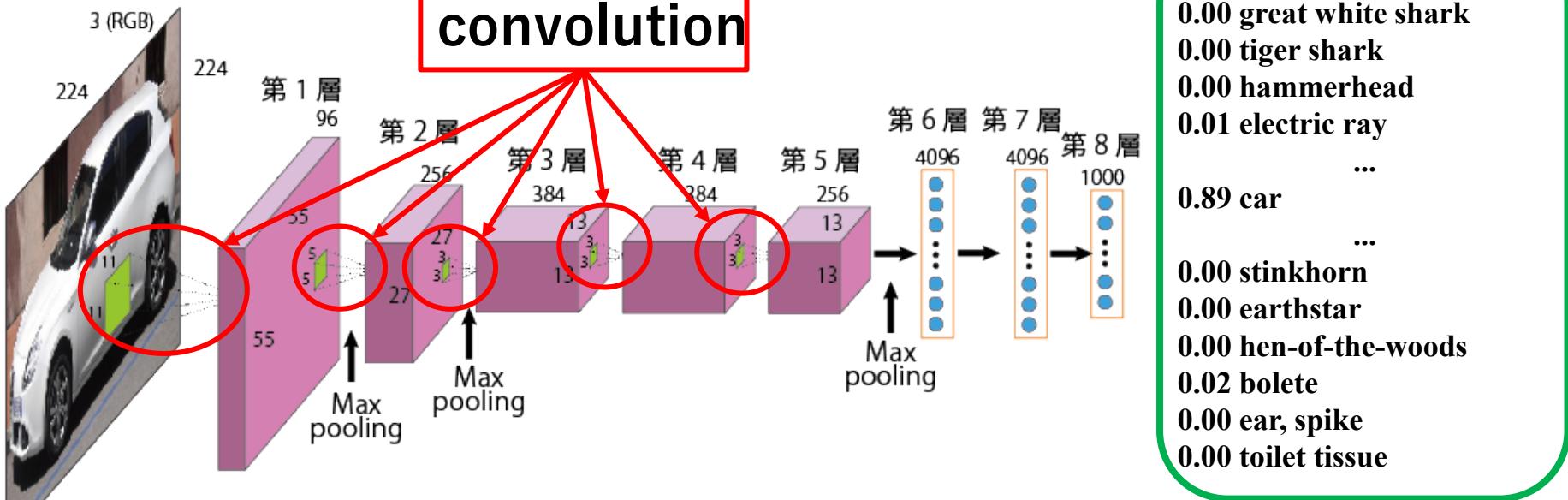
ref. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS2012

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0

https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

畠み込みニューラルネットワークにおける畠み込み演算

- 第1層から第5層までは、2次元デジタルフィルタで説明した畠み込み演算(convolution)を行っている
- ただし、2次元デジタルフィルタではフィルタは人がデザインしていたのに対し、CNNではデータから学習により取得する



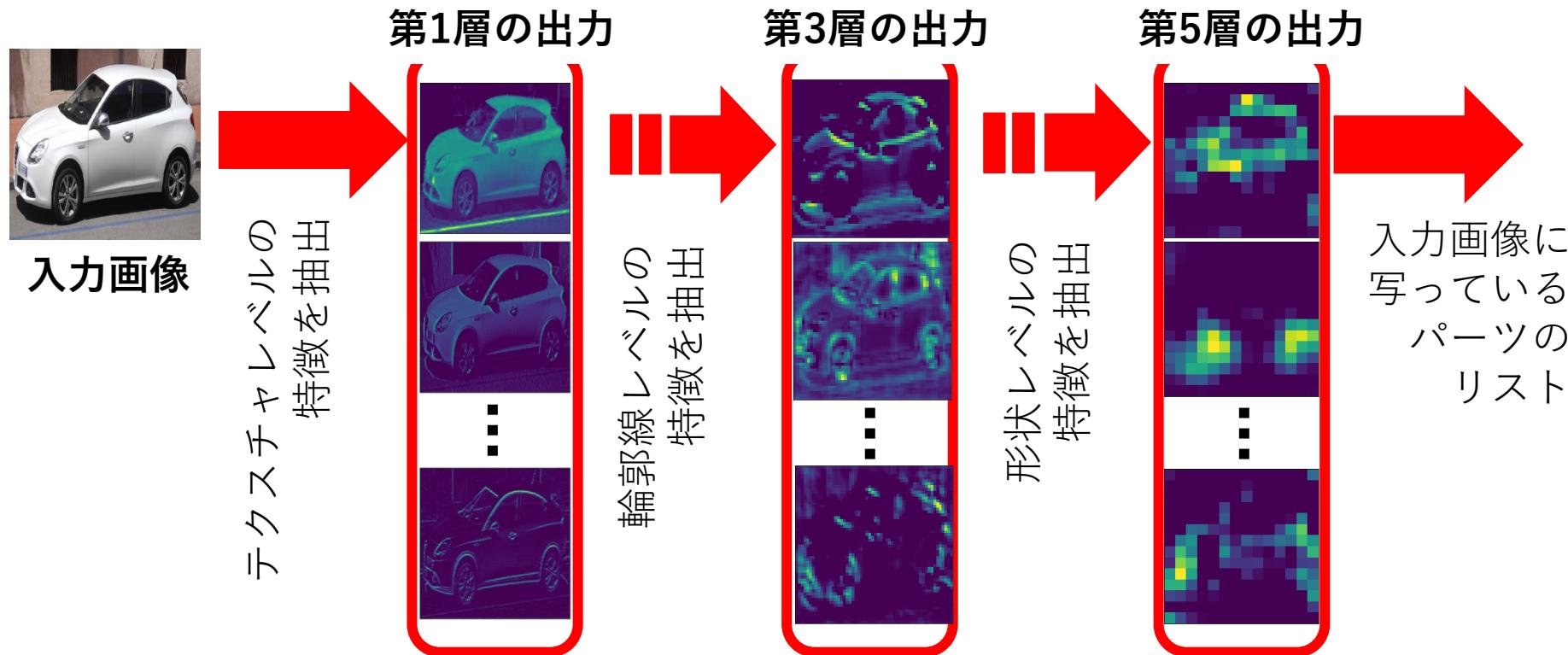
ref. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS2012

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0

https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

CNNにおける画像のフィルタリング

- 層を経るごとにより具体的な形状の特徴を取り出していく
- 第5層になると、「入力画像にどんなパーツが写りこんでいるか（実際には尤度分布）」がわかってくる
- 「入力画像に写っているパーツのセット」が「他のクラスに比べ、車にありがちなパーツのセット」であるならば「車」と判別



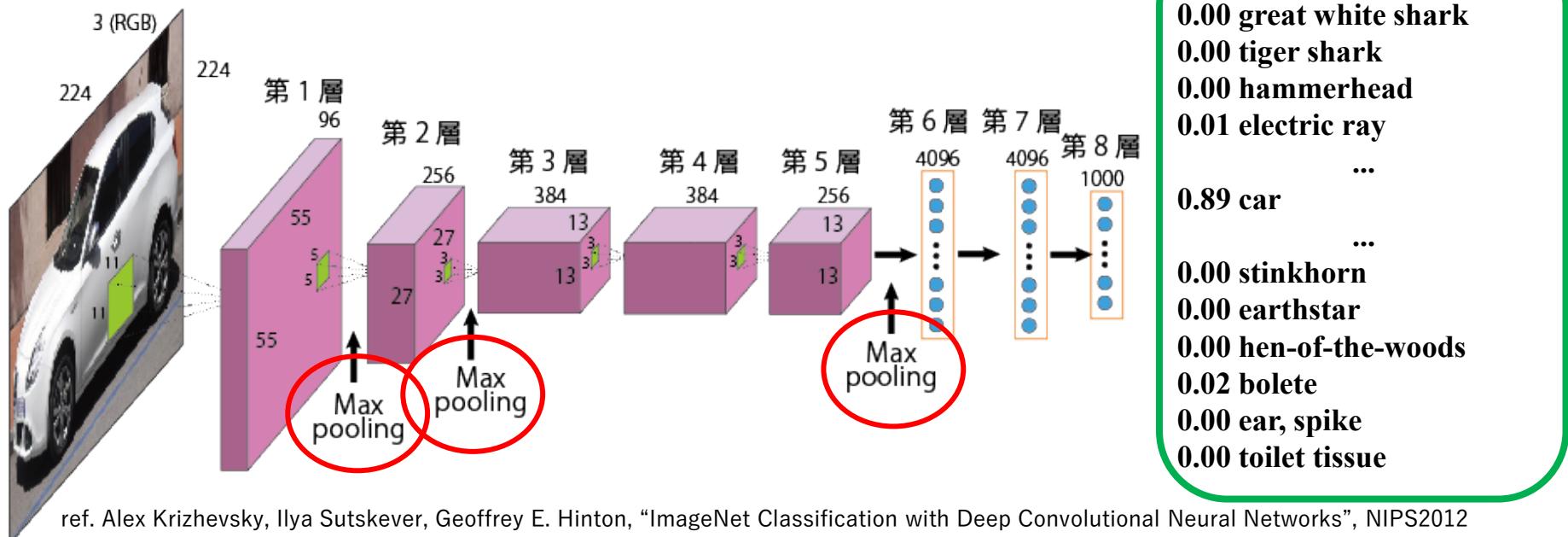
ref. (2020/4/3): Wikimedia commons:

File:Red Apple.jpg CC BY 2.0

https://commons.wikimedia.org/wiki/File:Red_Apple.jpg 東京大学 数理・情報教育研究センター 山肩洋子 2021 CC BY-NC-SA

畳み込みニューラルネットワークにおける プーリング (pooling)

- ・ 画面を小さく区切り、各区間ごとに画素をまとめて1つの値にする
 - ・ データのサイズを小さくする役割
 - ・ 物体が写っている位置や角度の違いに頑健にする役割
- ・ 最大値を取る場合 max pooling、平均値をとる場合 average poolingと呼ぶ



ref. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", NIPS2012

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0

https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

深層学習はそれまでの画像処理と何が違ったのか？

- AlexNet (2012)登場より前の画像認識では、CNNにおける第1~2層の出力に相当する情報を使って認識していた
→ Deep Networkと対比してShallow networkと呼ばれる
- 2000年前後にいくつかの技術革新
 - 数学的解法：勾配消失問題（層を深くすると学習が進まなくなる現象）に対する効率的な解決法の提案（1990年代後半）
 - GPGPUの発展：コンピュータグラフィックスの描画に用いられていたGPUをベクトル計算機とみなして気象や地震シミュレーション等、数値計算に利用（2006年NVIDIAがCUDA提供開始）
 - Big Data時代の到来：
Webで画像やテキストなどが大量に収集できるようになり、モデルの学習に使えるデータが爆発的に増加
- 様々な画像処理タスクに対する学習データセットが公開
 - 学習データを使わない自己教師あり学習 (self-supervised learning) の研究も進められている

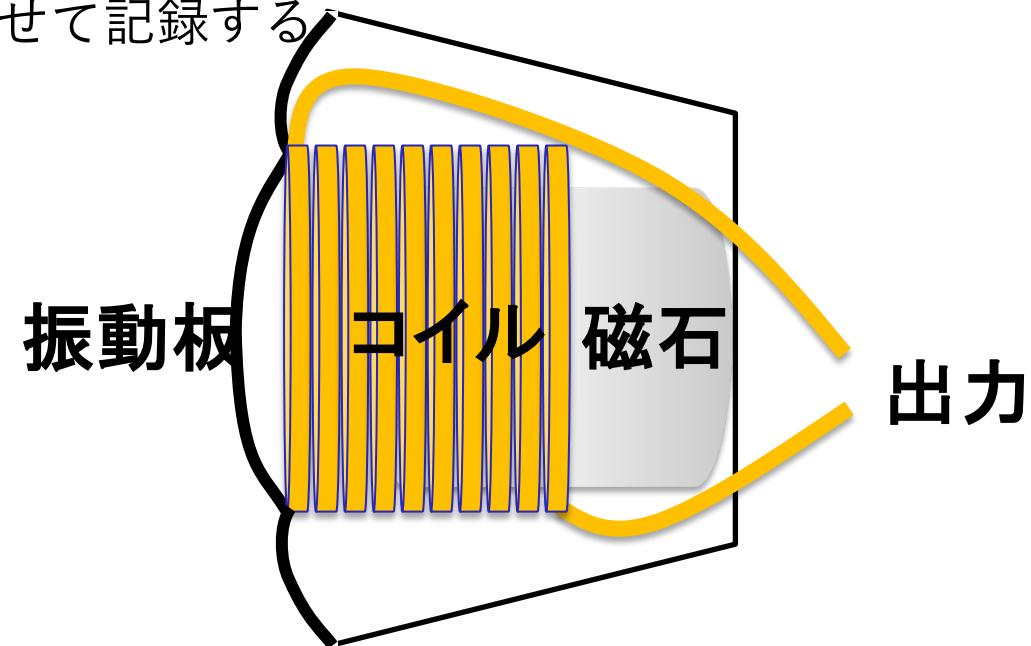
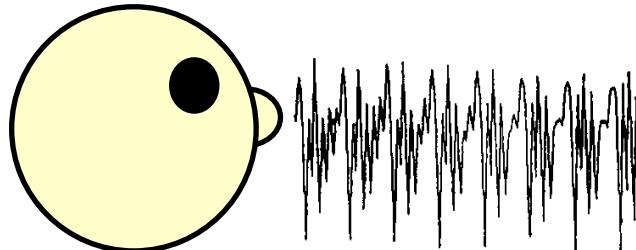
4. 音響処理・音声認識技術

4.1 デジタル音の表現

音声の収録

—ダイナミックマイク（ムービングコイル型）の原理—

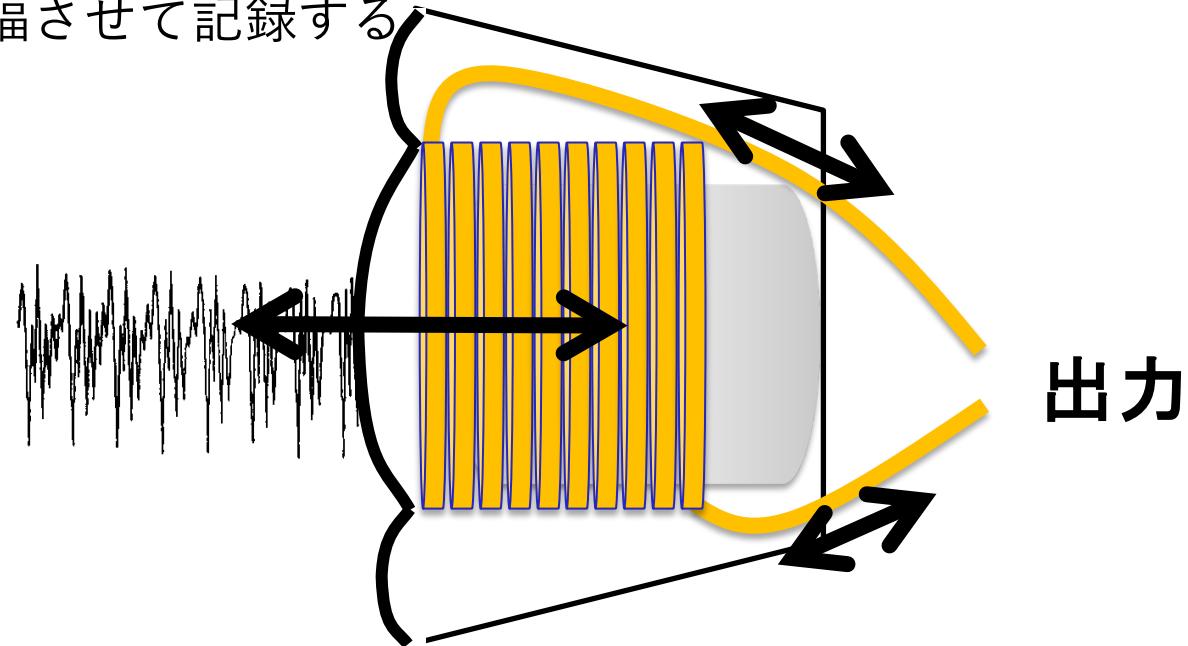
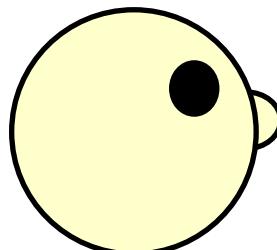
- ・ 振動板に音波（粗密波）が当たると振動板が揺れる
- ・ 振動板が揺れるとコイルが揺れる
- ・ 磁界に対してコイルが動くと電流が流れる
→この微弱な電流を增幅させて記録する



音声の収録

—ダイナミックマイク（ムービングコイル型）の原理—

- ・ 振動板に音波（粗密波）が当たると振動板が揺れる
- ・ 振動板が揺れるとコイルが揺れる
- ・ 磁界に対してコイルが動くと電流が流れる
→この微弱な電流を增幅させて記録する



プロ用の収録機器では周波数特性に優れたコンデンサマイクが使われる

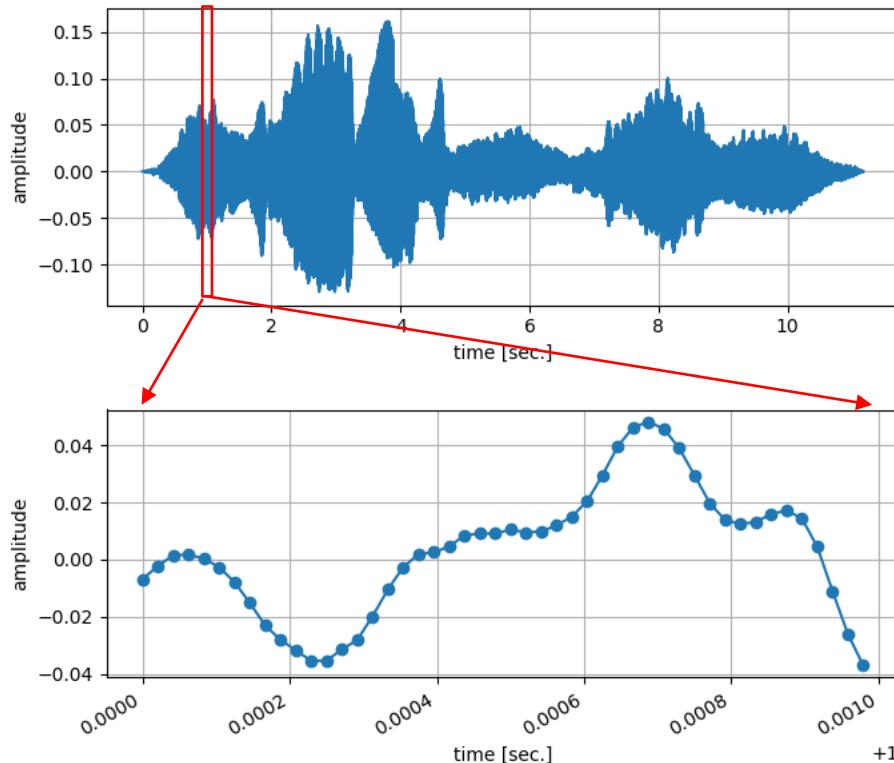
デジタル変換後の音声波形

時間方向の離散化：

波形を1秒間に何回記録するか→サンプリングレート

振幅方向の離散化：

振幅方向の揺れの大きさを何階調で記録するか→ビット深度



0.001秒分の波形を
拡大すると…

アナログ-デジタル変換 (AD変換)

標本化：サンプリングレート、フレームレート

- 時間的に連続する波形を一定間隔ごとに記録する
- 単位時間あたり何回記録するか→サンプリングレート（単位はHz）

量子化：ビット深度、量子化ビット

- 1サンプルを何ビットで記録するか？
- 振幅方向の階調の粒度に影響
例) 1サンプルあたり16bitで表現するなら、 $2^{16}=65536$ 階調
ただし、振幅は正と負があるため、正負それぞれの階調はその半分

音に限らず、AD変換する際は常に生じる問題！

サンプリング定理

サンプリングレートが $F\text{Hz}$ のとき $f\text{Hz}$ の波は $(F-f)\text{ Hz}$ の波と区別できない

サンプリングレートが $F\text{Hz}$ のとき、 $F/2$ よりも高い周波数の波 f を収録してしまうと、それは $(F-f)$ の周波数の波と混ざって記録される
→ 折り返し雑音 (aliasing noise)と呼ぶ

逆に言えば・・・

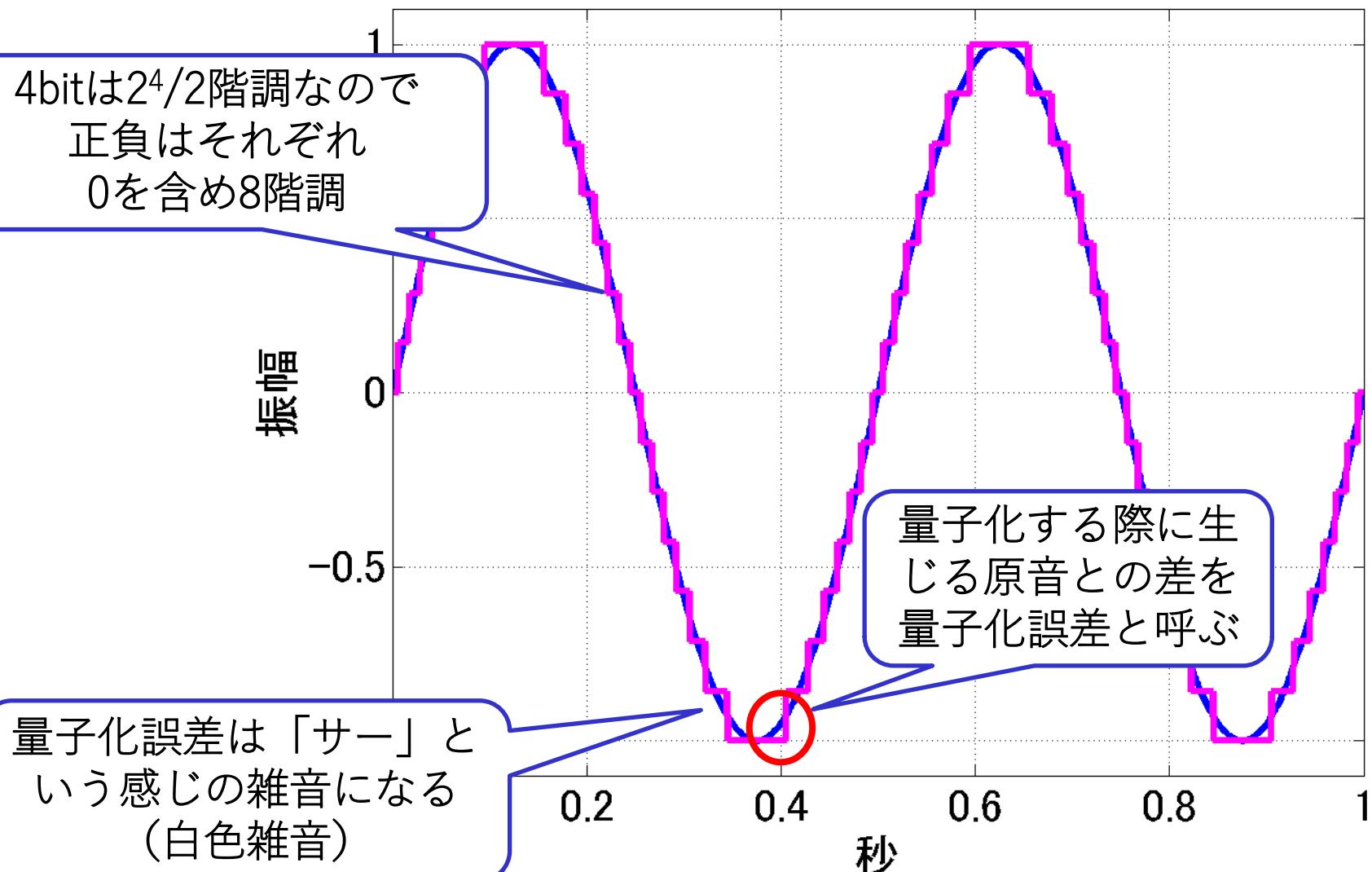
収録対象に含まれている音（通常は様々な周波数の波が混ざった混合音）の最大周波数が $f_{max}\text{ Hz}$ である場合、これを記録するためには $F > 2f_{max}$ であるようなサンプリングレート F で記録しなければならない！

→ サンプリング定理

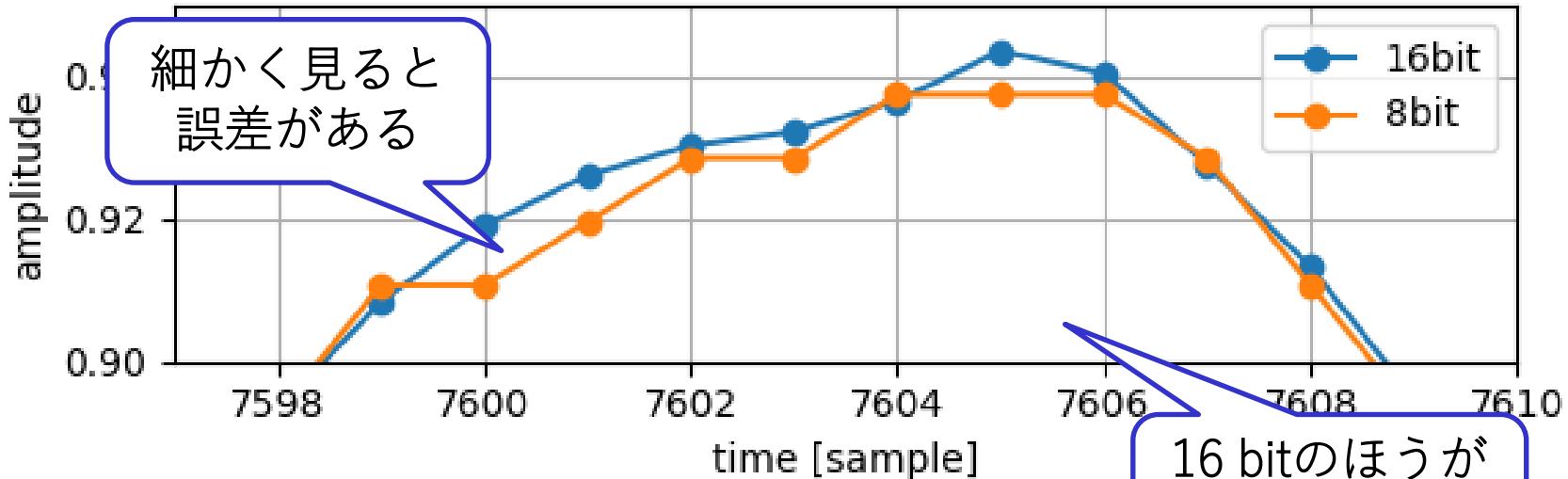
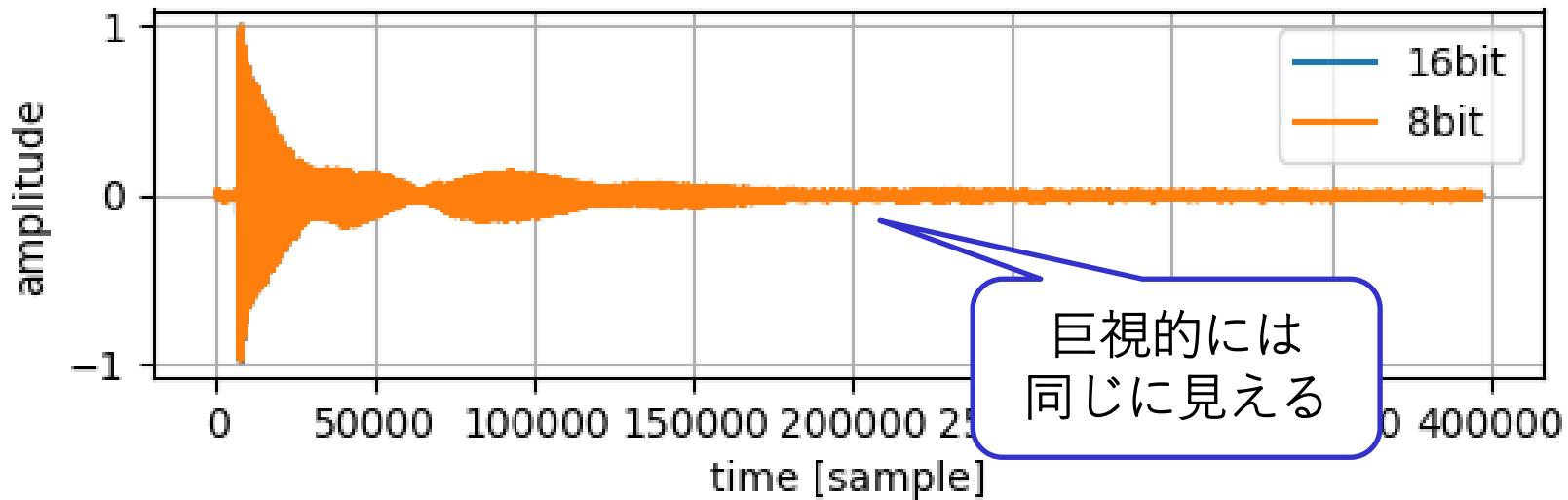
サンプリングレートの $1/2$ の周波数をナイキスト周波数と呼ぶ

量子化と量子化雑音

ビット深度4 bitで量子化するとすると…



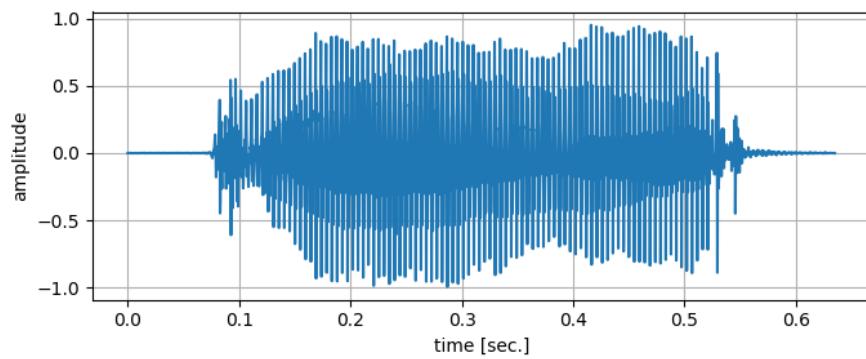
実際の波形でビット深度が8-bitと16-bitを比較



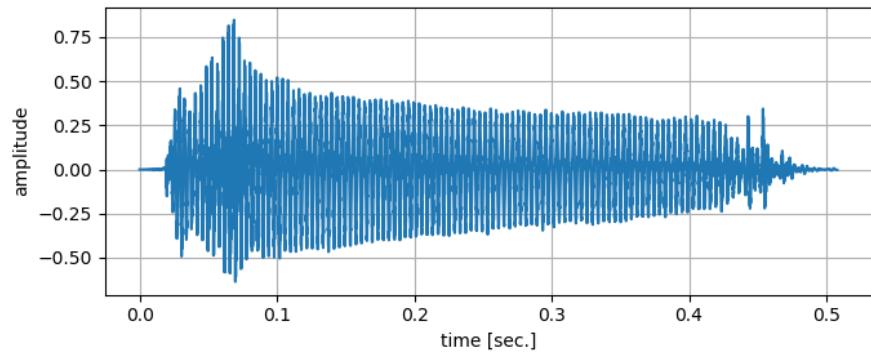
4. 音響処理・音声認識技術

4.2 周波数分解

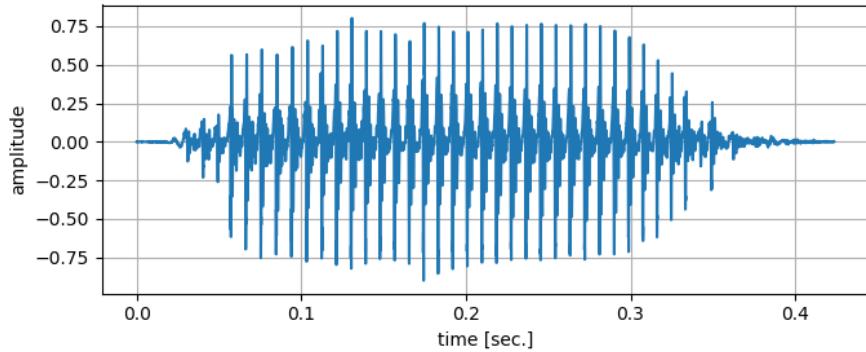
音声波形の特徴



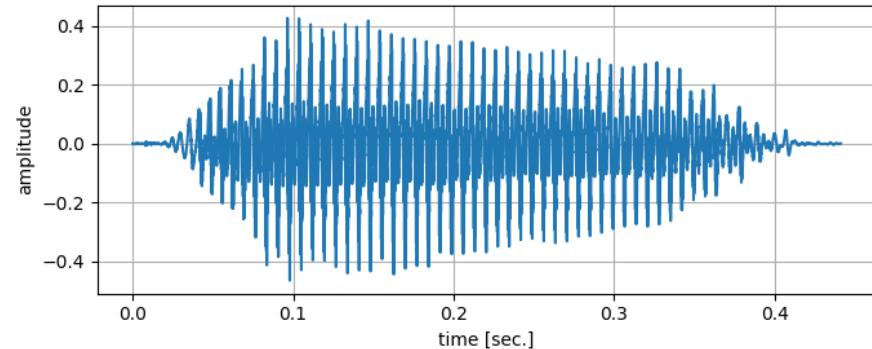
女性音声「あ」



女性音声「い」



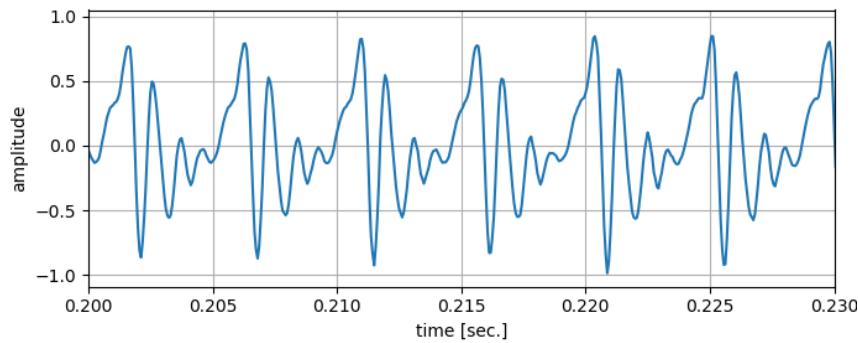
男性音声「あ」



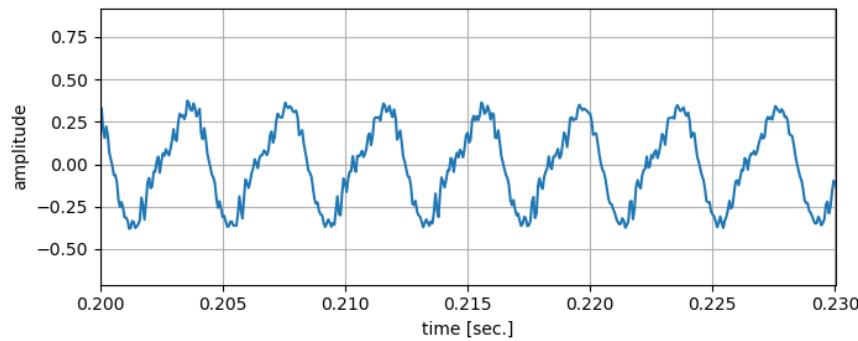
男性音声「い」

音声認識では左の波形を「あ」、右の波形を「い」と判別できます
どうやって見分けているのでしょうか？

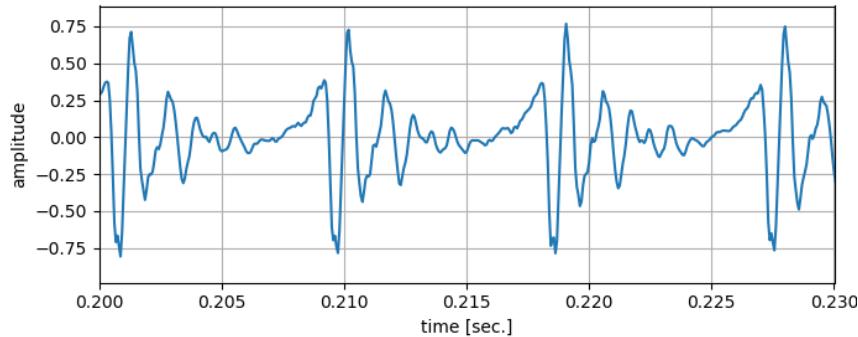
音声波形の特徴（拡大）



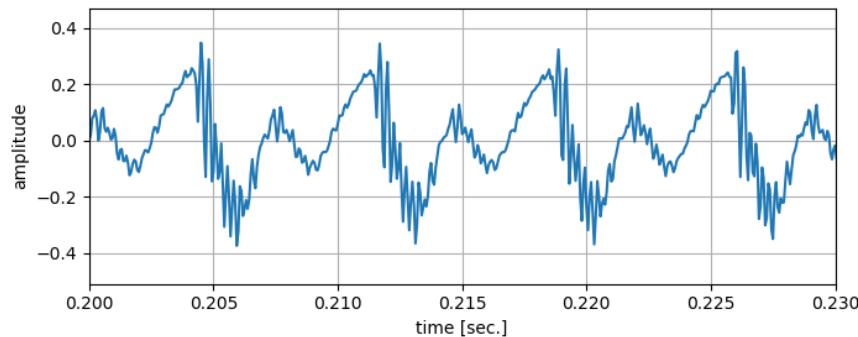
女性音声「あ」



女性音声「い」



男性音声「あ」



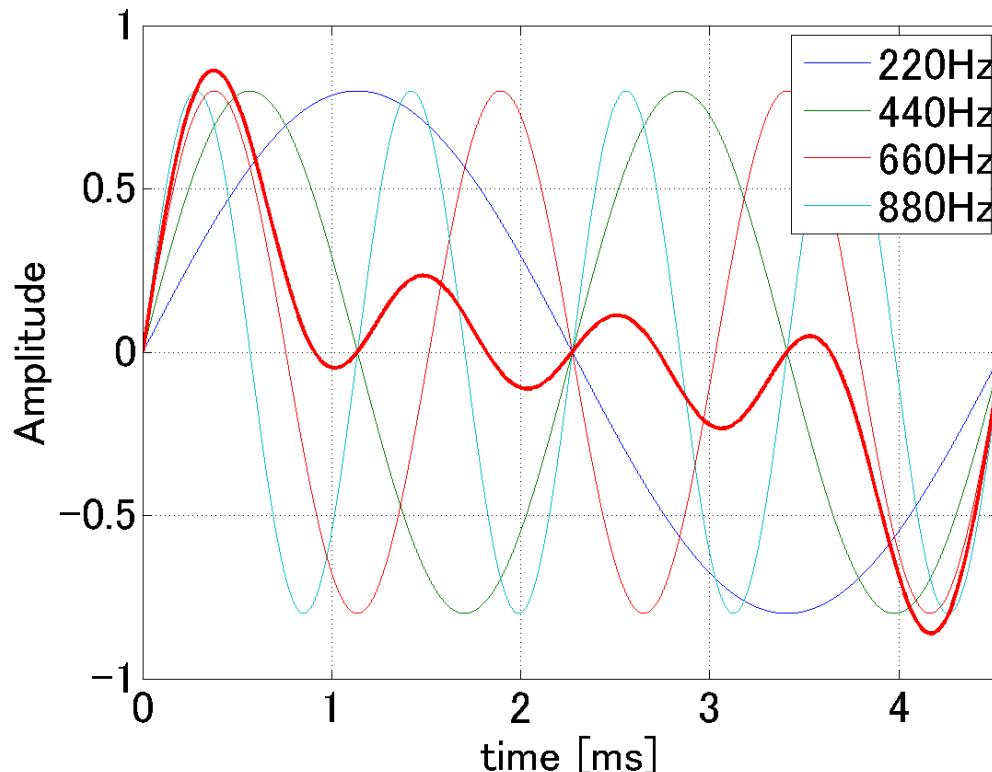
男性音声「い」

拡大するとなんとなく波形は似て見えますが…

音の高さごとに波を分解してみましょう！→ 周波数分解

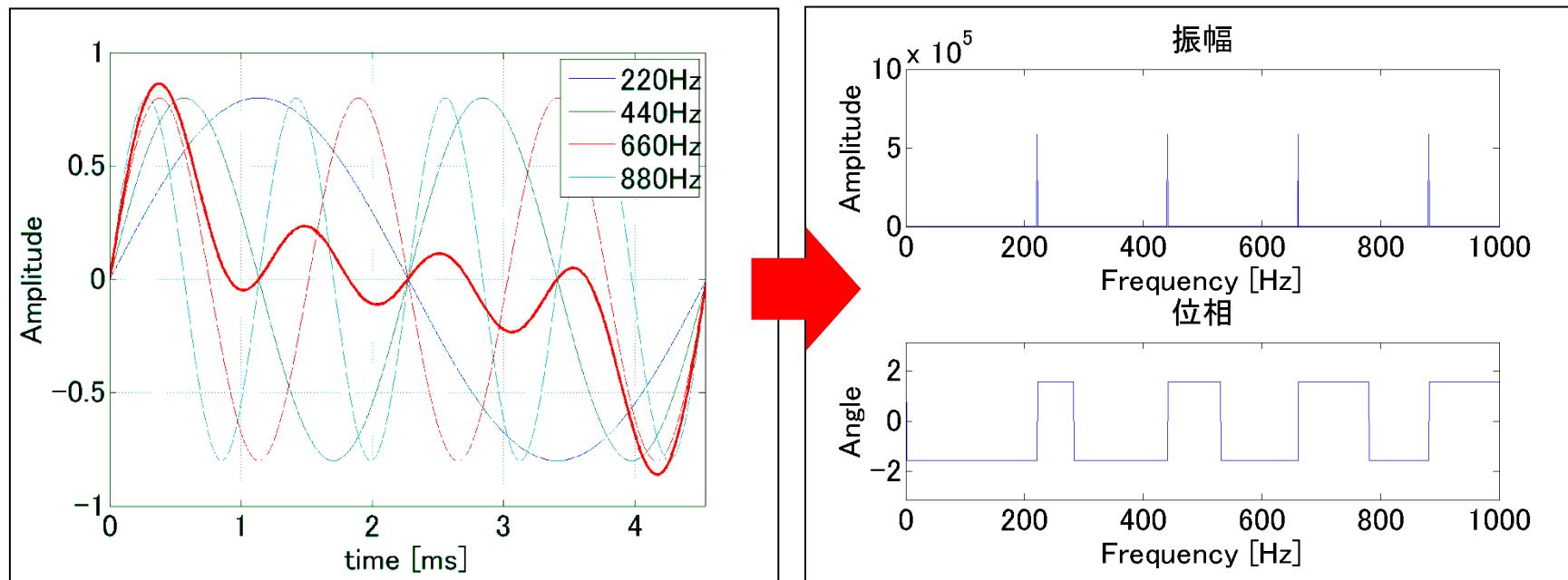
音は正弦波の重ね合わせとみなすことができる

あらゆる音は振幅・位相・周波数（波長）によって
決まる正弦波（純音）を重ね合わせたものと考えることができる



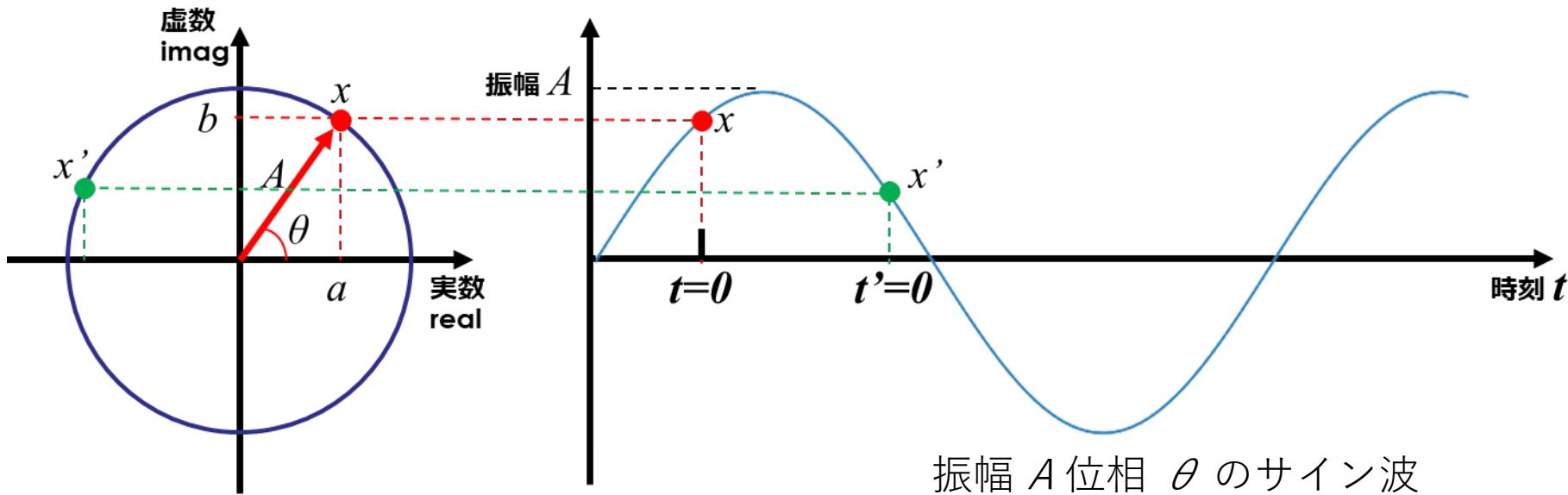
周波数スペクトル

- 音を周波数ごとに分解して、その各周波数の振幅や位相で特徴を表現
→ 周波数スペクトルと呼ぶ
- デジタル音は離散フーリエ変換により各周波数成分に分解可能
 - 実際にはその高速化手法である高速フーリエ変換（FFT）が使われる



正弦波の振幅と位相およびその複素数表現

- ある純音は周波数・振幅・位相の組み合わせで一意に定まる
- 振幅と位相は複素数で表現されることがある
 - 下の図は振幅が A 、位相が θ の波形
 - 時刻 $t=0$ のときの波の値 $x = a + bi$
 - このとき、振幅は $A = \sqrt{a^2 + b^2}$ 、位相は $\theta = \tan^{-1} b/a$
ここで $\tan^{-1}(x)$ とは $\tan(y) = x$ となるような角度 y のこと

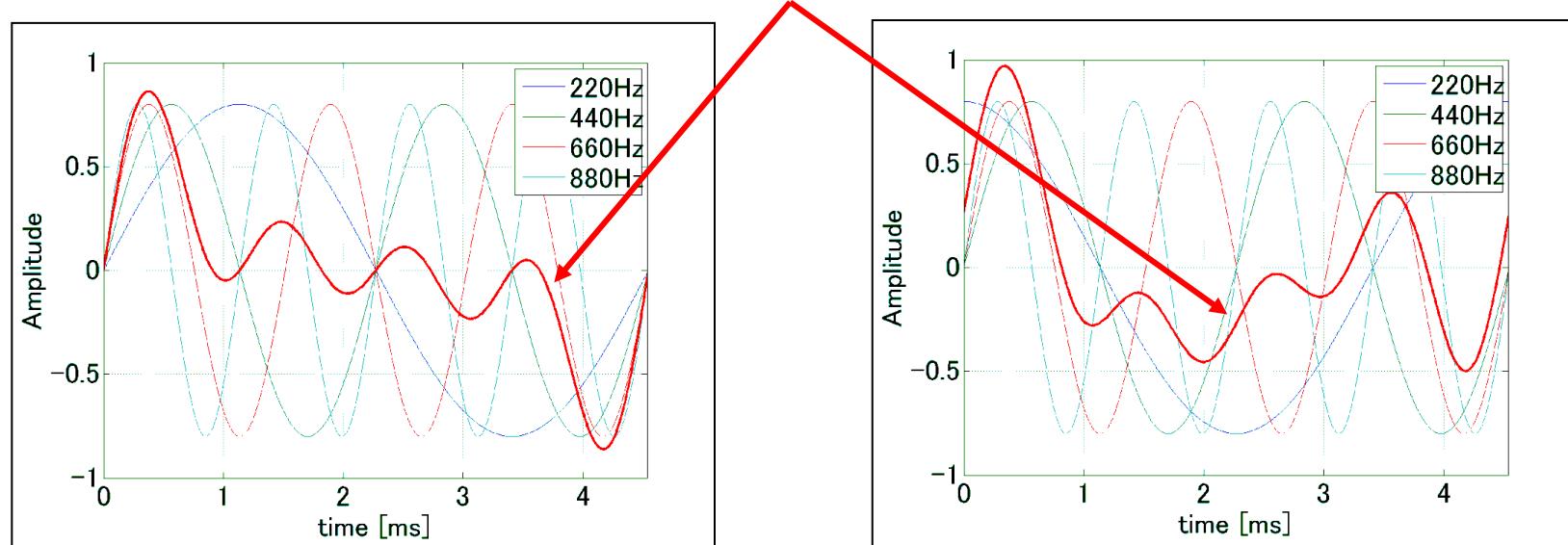


聴覚の特性

複数の波長の音を合成した複合音では、いずれかの波長の音の位相が変化しても、聴覚はその違いを知覚できない

→ それでも人間は音声を聞き分けられているのだから、
音声認識でも振幅成分だけ見ればいい

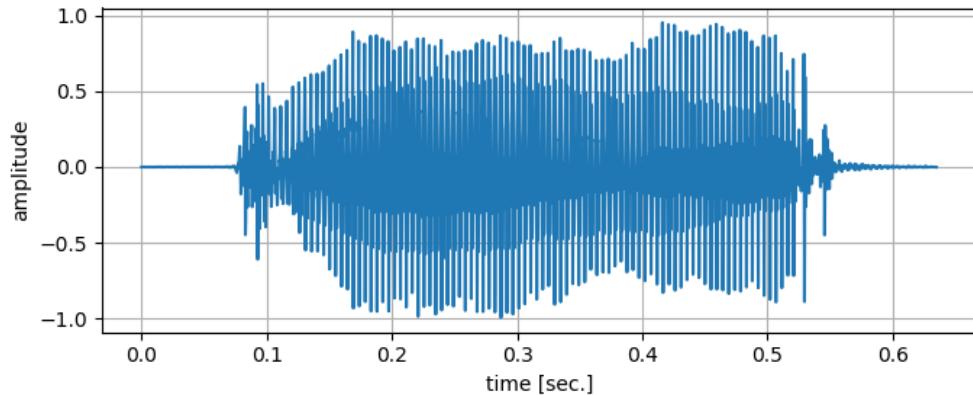
これら二つの複合音は、波形は異なるが、
これらを聞き比べても人間の耳は区別ができない



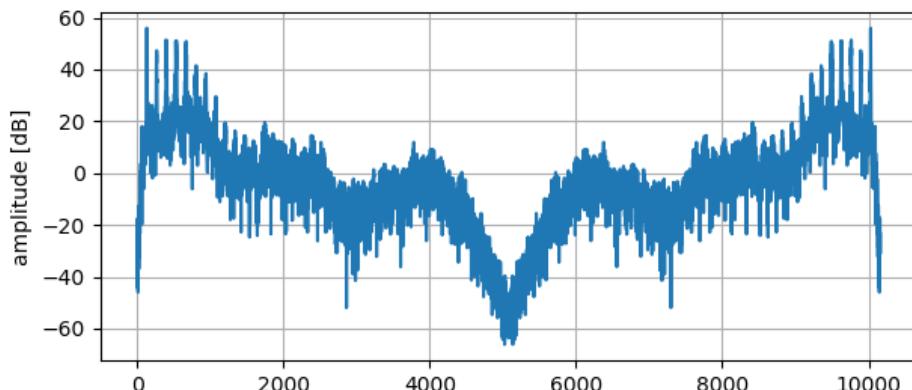
220Hzの波の位相が
左のグラフより90度ずれている

音声の周波数分析手順 (1/2)

女性音声「あ」の波形



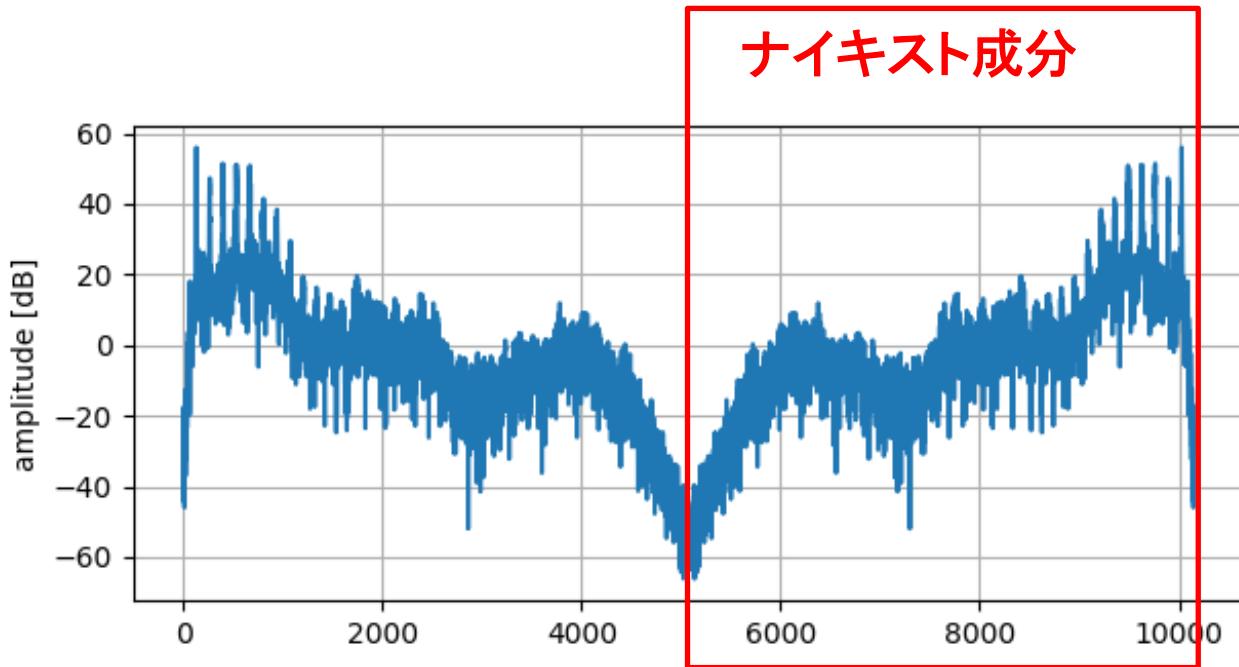
- 高速フーリエ変換 (Fast Fourier Transform, FFT)
- 振幅成分は周波数成分を X とすると $20\log_{10}(X)$ としてdB値に変換



注：元の波形のフレーム数が
10161であるため、
周波数成分も10161次元となる。

音声の周波数分析手順 (2/2)

中央から上の周波数成分はナイキスト成分であるため除去する
(参考 : サンプリング定理)



注：周波数成分が10161のため、5081番目のフレームから10161番目のフレームまでを削除

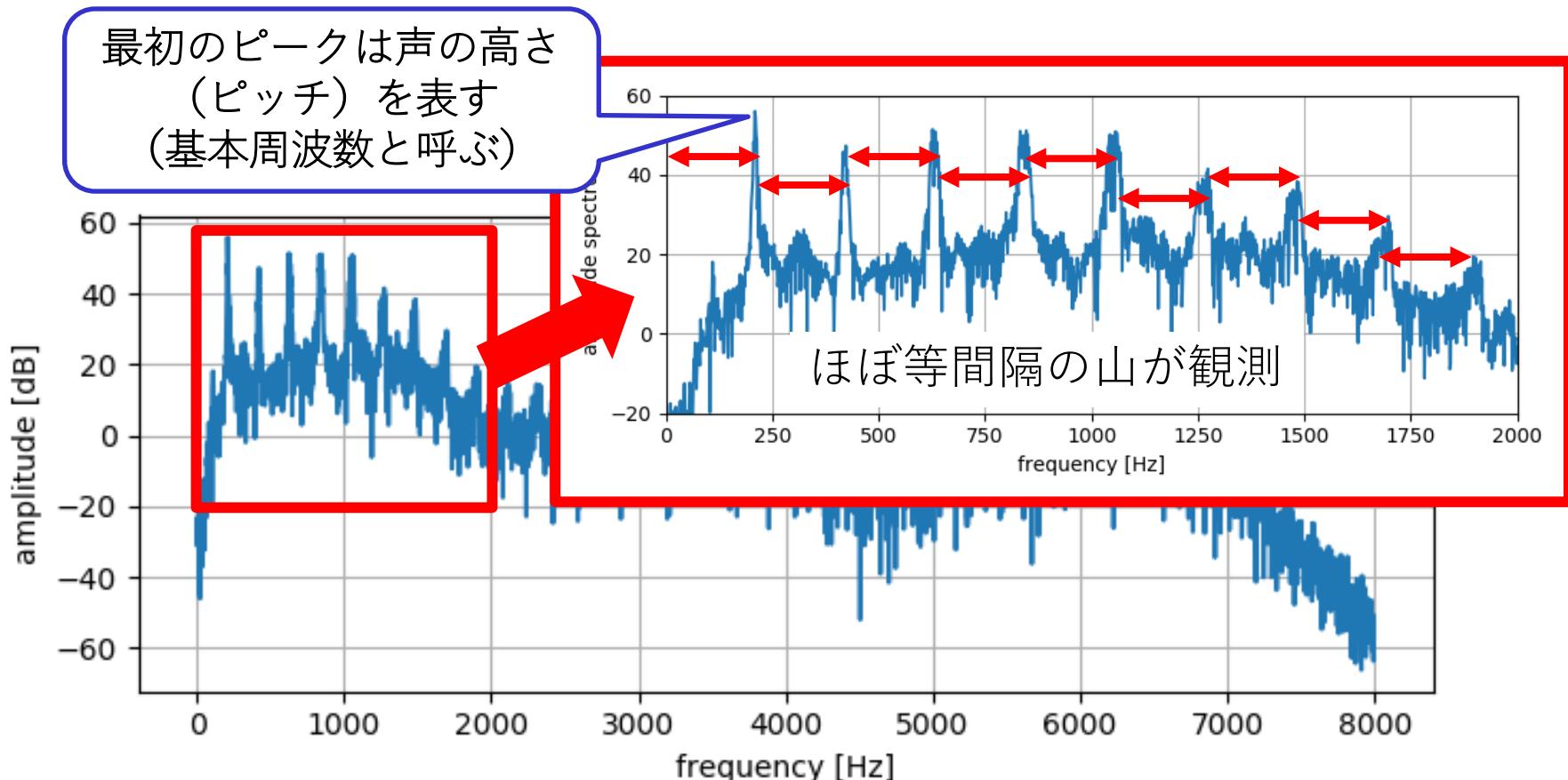
この音声ファイルのサンプリングレートは16kHz
→ 8kHz以上はナイキスト成分なので無視する

音声の周波数成分の特徴

特に低周波領域におよそ一定間隔の山が見える

→ 主に声帯で生じた倍音成分

楽器の単音でも同じおよそ一定間隔の波が観測できる

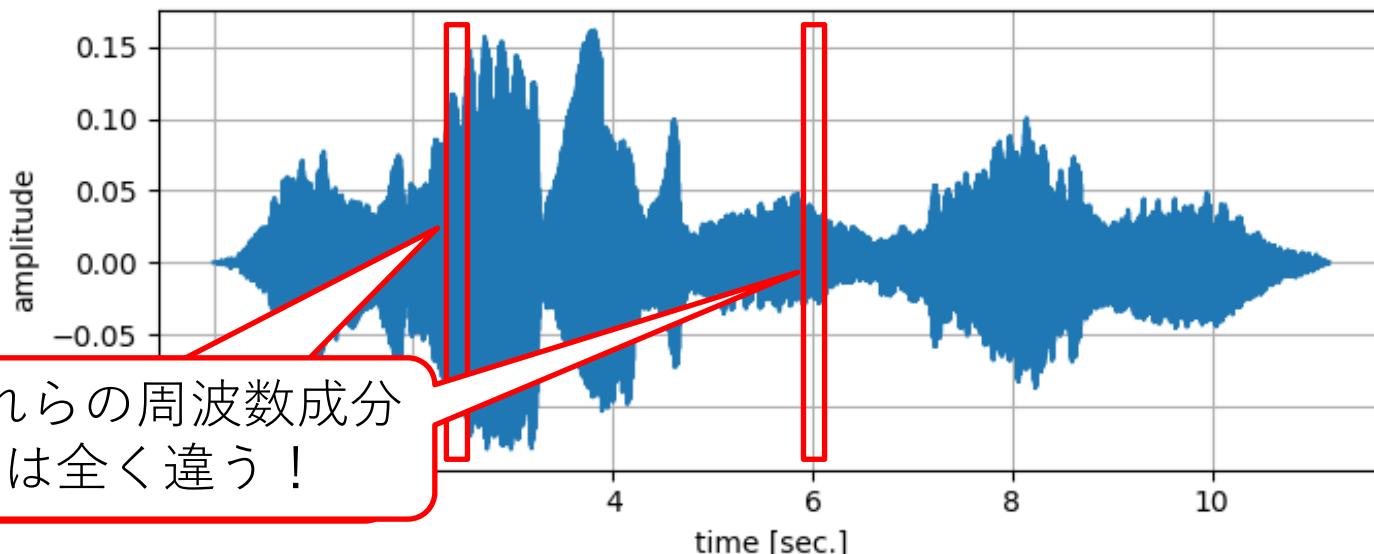


時間的に変化する音の周波数分解

通常の音声や音楽は時間的に周波数成分が変化
しかしフーリエ変換は

- 一定区間の音声を入力として周波数分解
- ある瞬間の音（つまり1サンプル）では周波数分解できない

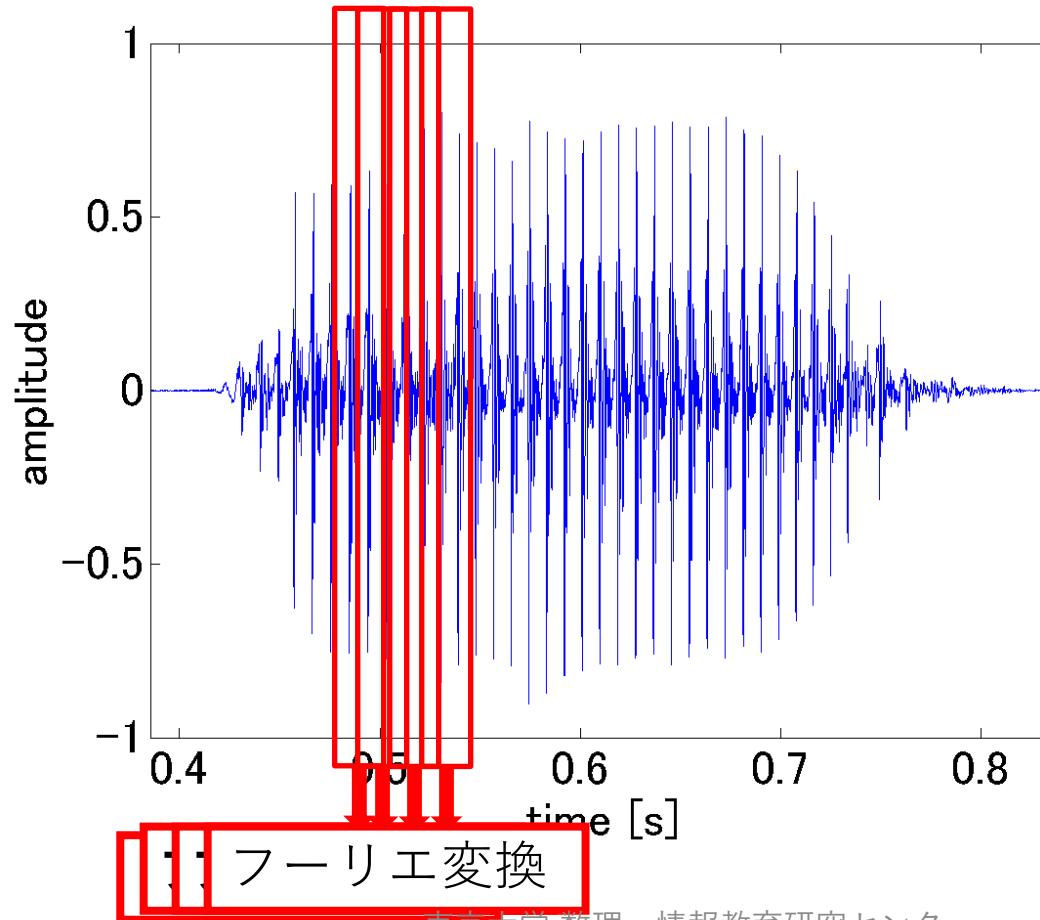
時間的に変化する音はどのように周波数分解したらしいか？

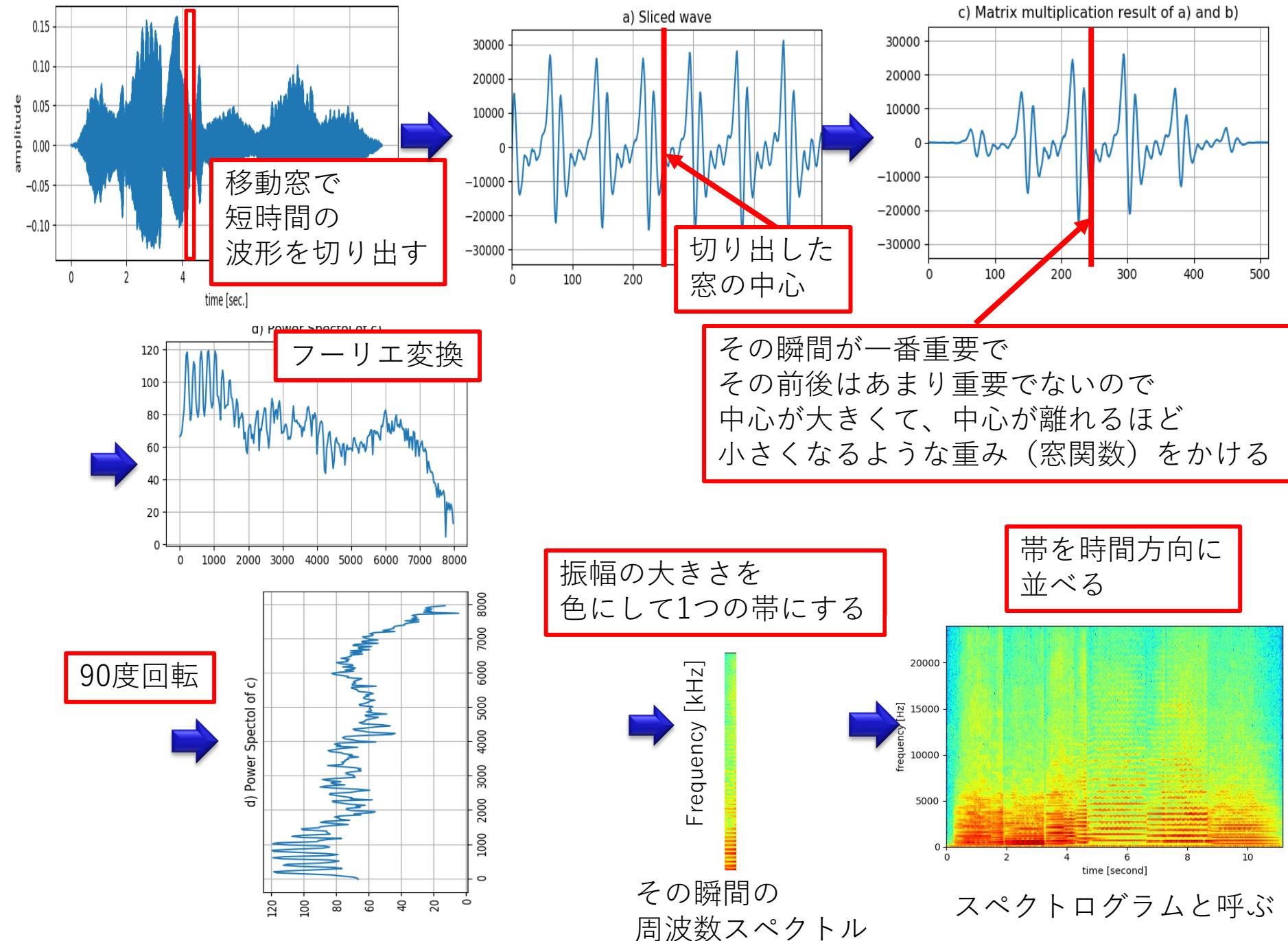


移動窓による短時間音声の切り出し

波形を短い区間で切り出して周波数分解

周波数成分を時間方向に並べればいい！

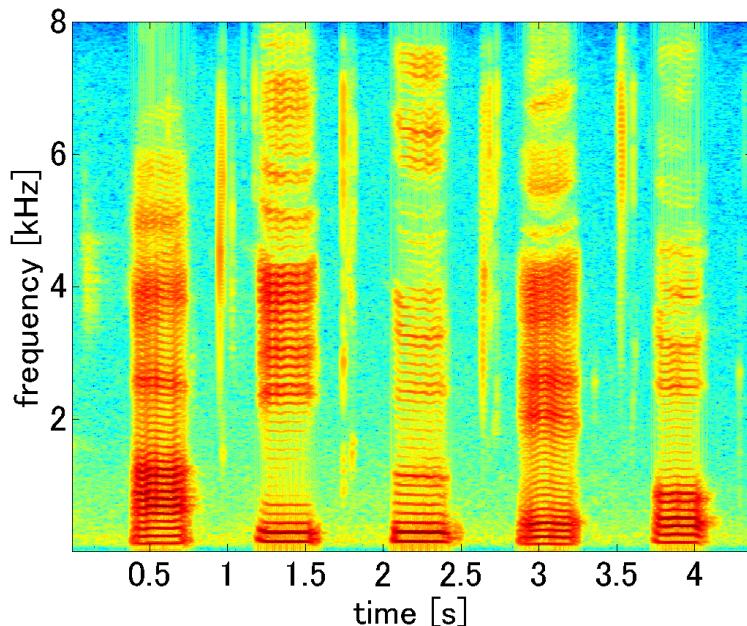




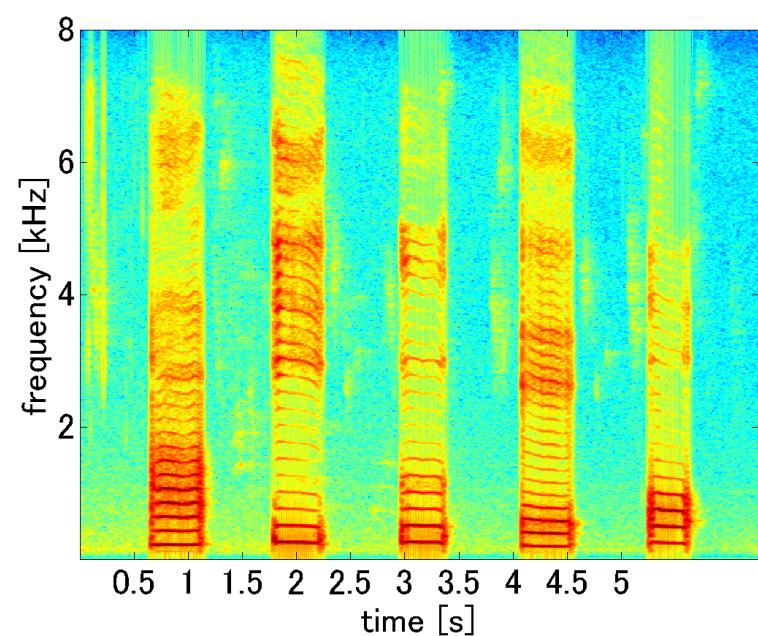
音声のスペクトログラム

「あ・い・う・え・お」と発音したときの男性と女性の音声のスペクトログラム

- 声の高さの違いによって縞模様の幅が違う
(男性の方が声が低い = 基本周波数が低い = 倍音の周波数間隔が細かい)
- 男性と女性で同じひらがなを発音している部分を見比べると
→ 縞の間隔を無視すれば柄が似ている → 音声認識の手がかり



男性音声

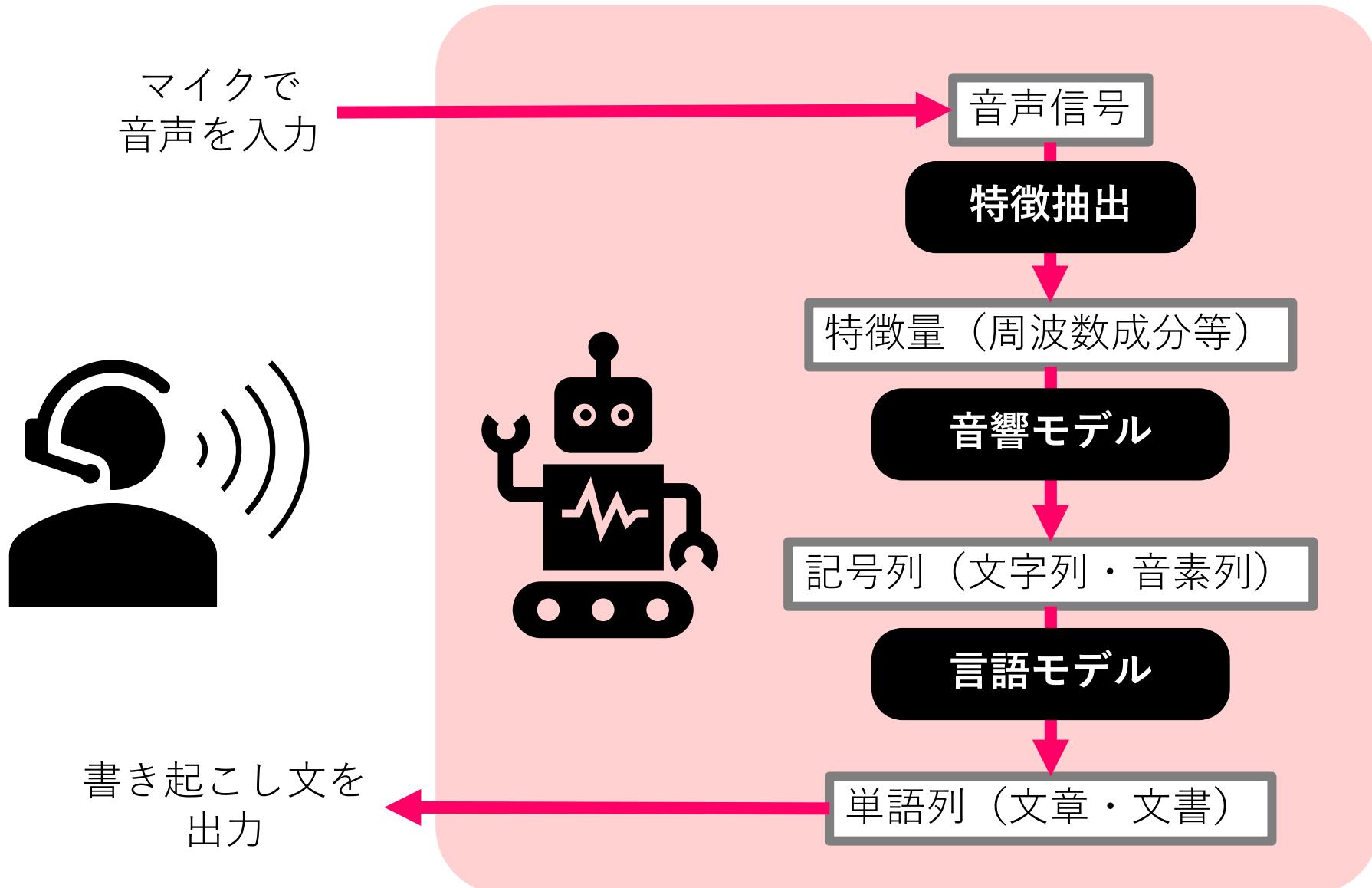


女性音声

4. 音響処理・音声認識技術

4.3 音声認識

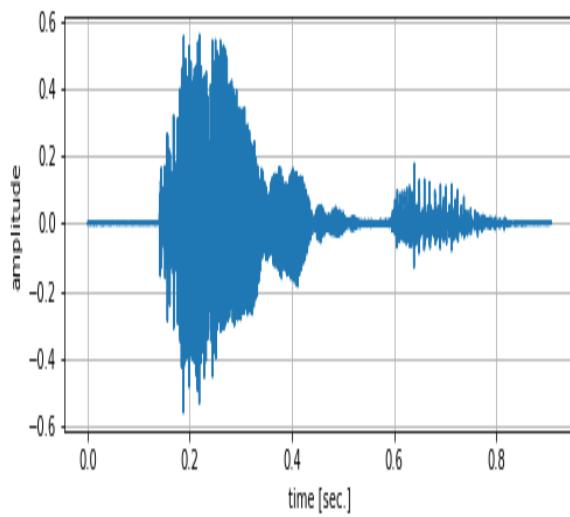
音声認識の流れ



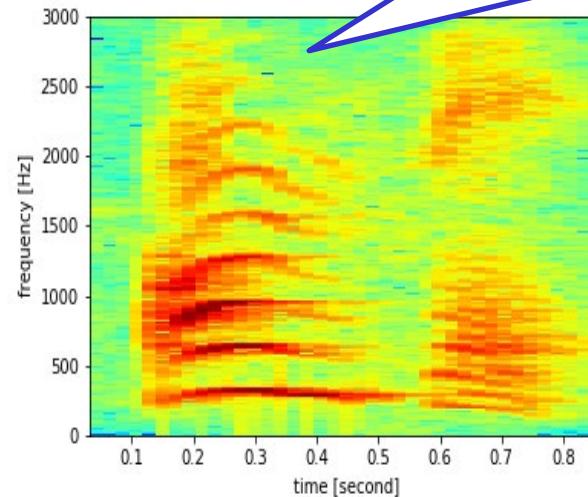
特徴抽出

音声波形を周波数解析により対数パワースペクトルに変換

- 音声波形のある区間を切り出したものを256次元のベクトルに変換
- 時間方向に音響特徴量が並んだ時系列データとなる



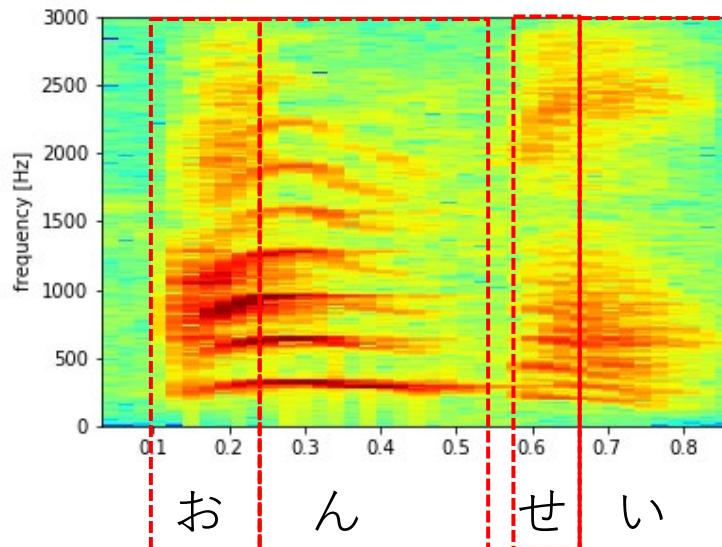
周波数
解析

時間方向に並んでいる

音響モデルの課題

- 音響モデルは音響特徴量の系列をサブワード（文字列や音素列等）の系列に対応付ける処理
- 通常、音響特徴量系列はサブワード系列よりもはるかに長い
 - “おんせい”を発話→“お”に対応するフレームは1つではない
 - 同じ“お”を発音した場合でも、ゆっくり話せばそれだけ多くの音声フレームが生成される
 - 長さの異なる系列どうしをどうやって対応付けるか？



周波数スペクトルは
時間方向に連続的に変化
している点に注意

音響モデルの発展

- GMM-HMM (1980年代から2010年ごろまでこの手法が主流)
 - 音声フレームの冗長性を隠れマルコフモデル (Hidden Markov Model; HMM)と呼ばれる状態遷移モデルでモデル化
 - HMMの各状態における音声の生成確率を混合正規分布 (Gaussian Mixture Model; GMM) でモデル化
 - 単語に対し、その音素列（どのような発音かを記号であらわしたもの）を記した「発音辞書」が必要
- DNN-HMM
 - GMM-HMMのGMMを多層ニューラルネットワーク (Deep Neural-Network; DNN)に置き換えたもの
 - GMM-HMMを前提に開発されてきたリソースがそのまま使えるため、製品ではDNN-HMMが主流で、引き続き改良が続けられている
- End-to-Endモデル
 - 音声特徴を入力とし記号列（文字列や音素列）を出力するモデルを1つのニューラルネットワークで構成
 - HMMで必要だった「発音辞書」が不要、モデルが1つなため学習が単純
 - 代表的なモデルはConnectionist Temporal Classification (CTC)

音響モデル：CTCの枠組み

サブワード系列 L

('_'は空白を表す特殊文字)

n i c e _ t o _ m e e t _ y o u

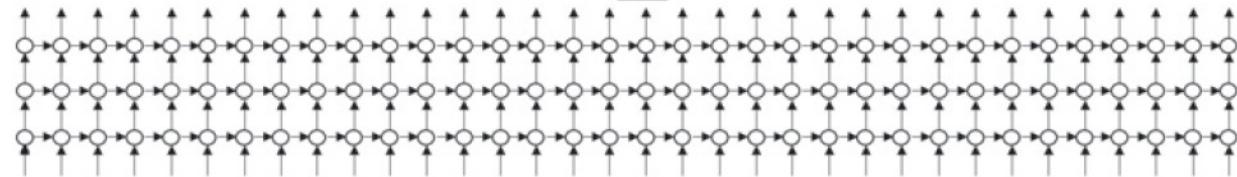
↑ Collapsing 関数 $\psi()$

ラベル系列

$C = \{c_1, \dots, c_T\}$

$\begin{cases} \emptyset n \emptyset i c e \emptyset_t t \emptyset o \emptyset_ \emptyset m m e \emptyset e e t \emptyset_ \emptyset y o o \emptyset \emptyset u u \emptyset \emptyset \\ \emptyset \emptyset n i c \emptyset e \emptyset_ t o o \emptyset_ \emptyset \emptyset m e \emptyset e t \emptyset_ \emptyset y y o \emptyset u \emptyset \emptyset \emptyset \emptyset \end{cases}$

ニューラル
ネットワーク



音響特徴量系列

$X = \{x_1, \dots, x_T\}$

↑

↑ 特徴量抽出

音声

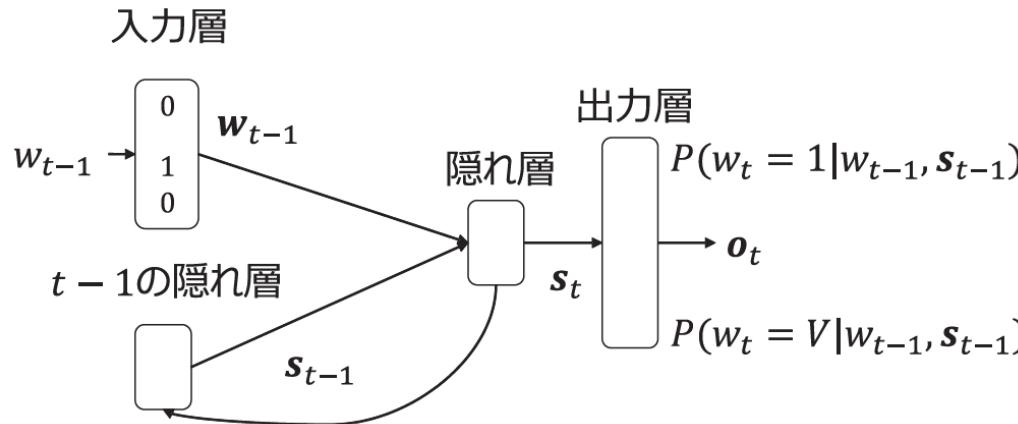


ref.神田 直之, “音声認識における深層学習に基づく音響モデル”, 日本音響学会小特集－音声言語処理における深層学習－, 図-4 CTC の枠組み, 73巻1号 p. 31-38, 2017.

言語モデルの課題

- 文を単語列と考え、単語列の起こりやすさ（生起確率）をモデル化
 - 音響モデルが文字や音素の列を出力 → 漢字を含む文に変換する処理を担当
 - 「今日は月曜」の生起確率は高く、「教派下津陽」の生起確率は低い
 - 「ぷろきし」→「プロキシ」「プロ騎士」どちらの単語が正しい?
→ 文脈を考慮して、より確率が高い方を選ぶ
- 最も基本的なモデル： n -gramモデル
 - ある単語の生起確率が、その直前の $n - 1$ 単語に（のみ）依存すると考える
例) 「ブラウザ の」の後ろであれば「プロキシ」
例) 「将棋 の」の後ろであれば「プロ騎士」
 - 直前ではなく離れた位置にある文脈を考慮することが難しい
- 再起型ニューラルネットワーク (Recurrent neural network; RNN)
 - 直前の $n - 1$ 単語ではなく、直前の1単語に加え、前の隠れ層の出力ベクトル（ここにそれまでの文脈情報が埋め込まれている）から単語を予測
 - LSTM (Long Short-Term Memory) や Attention を使うものもある
- 自然言語処理分野で開発されたTransformerを導入し、音声特徴から単語列までを一気に扱うモデルも研究されている

RNN言語モデルのモデル構造



単語列 $W = w_1, \dots, w_T$ の生起確率 $P(W)$ を次の式で計算

$$P(W) = \prod_{t=1}^T P(w_t | w_{t-1}, s_{t-1})$$

w_t : t 番目の単語の1-hotベクトル

w_t : t 番目の単語の単語ID (辞書の k 番目の語彙の場合 $w_t = k$)

s_t : t 番目の隠れ層の出力ベクトル

o_t : t 番目の単語の予測分布 (Softmaxの出力)

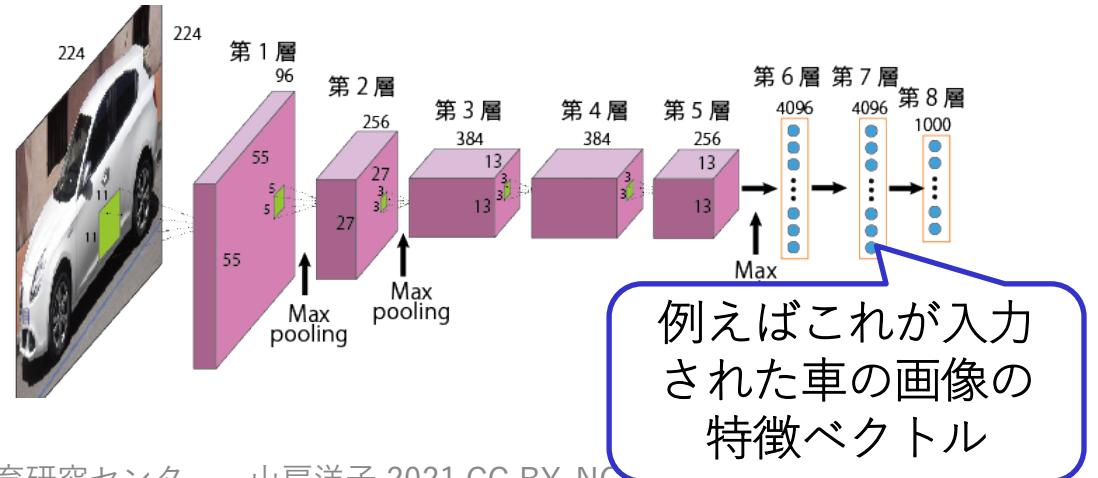
ベクトルはすべて、辞書に V 個の語彙が登録されているとき V 次元

ref.増村 亮, “深層学習に基づく言語モデルと音声言語理解”, 日本音響学会小特集－音声言語処理における深層学習－, 図-2
RNN言語モデルのモデル構造, 73卷1号 p. 39-46, 2017.

5. 認識技術の応用

表現学習・Feature Embedding

- 画像や音、単語や文などを特徴ベクトルに変換する処理
 - 特徴ベクトルに変換できれば、ベクトル間のコサイン類似度などによりサンプル間の類似度を計算できるため、画像検索等が実現できる
 - 画像や文書などに機械学習を適用する際の前処理としても行われる（意味のある情報を保持したまま次元圧縮することで計算コストを削減）
- 深層学習を使った手法が広く用いられている
分類問題などを学習したモデルに対し、サンプルを入力したとき、その中間層の出力をそのサンプルの特徴ベクトルとする
 - 中間層の出力は、最終層で正しく答えるための手がかりを持っている
 - 入力に近い層がより原始的な特徴（テクスチャや輪郭線等）、出力層に近い層がより具体的なパートを見分ける特徴を持つと言われる



ref. Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton,
"ImageNet Classification with Deep Convolutional Neural Networks", NIPS2012

ref. (2020/4/6): Wikimedia commons: File:" 10 Alfa Romeo Giulietta white Derivate cut.JPG CC0
https://commons.wikimedia.org/wiki/Category:CC-Zero?uselang=ja#/media/File:%22_10_Alfa_Romeo_Giulietta_white_Derivate_cut.JPG

Web APIを使った認識技術の活用

Web API (Web Application Programming Interface) とは？

- ・ サーバ上にあるソフトウェアの機能を、インターネットを介して外部のプログラムから利用できるようにする仕組み
- ・ ユーザが利用する端末の性能が脆弱な場合、高度な認識処理は実行できない場合がある
→ Web APIを使えば、端末はデータをサーバに送り、サーバ上で認識処理を行った後、結果だけを受け取ることができる
- ・ Google, Microsoft, 楽天、Amazon, Facebookなど多くの企業が、Web APIを介して自社の認識機能を提供している



本教材のまとめ

1. AIにおける認識とは	4
2. 認識技術の活用事例	1 2
- 画像認識の活用事例	1 2
- 音声認識の活用事例	2 4
3. 画像処理・認識技術	2 9
- デジタル画像の表現	2 9
- 二次元デジタルフィルタによる画像処理	3 6
- 深層学習による画像認識	4 8
4. 音響処理・音声認識技術	5 6
- デジタル音の表現	5 6
- 周波数解析	6 4
- 音声認識	7 8
5. 認識技術の応用	8 6