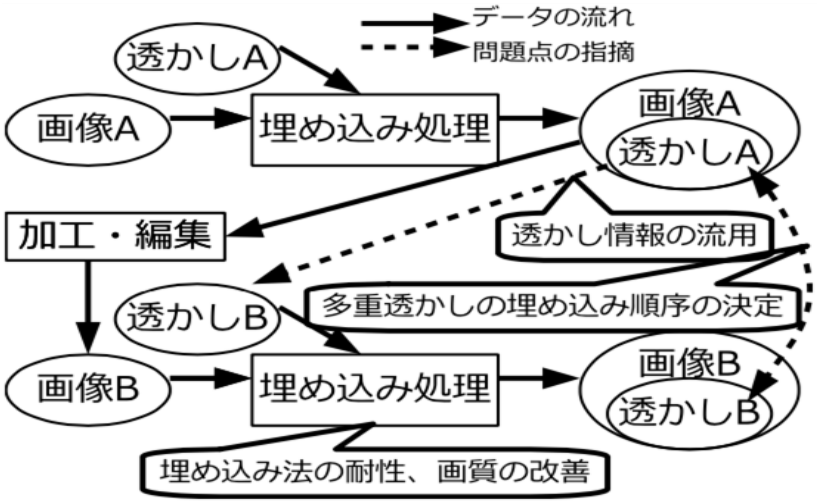
機械学習に基づいた
新しい知覚ハッシュ関数の構成と
電子透かしへの応用

神奈川大学　工学研究科　電気電子情報工学専攻
木下研究室　中谷憲　201870087

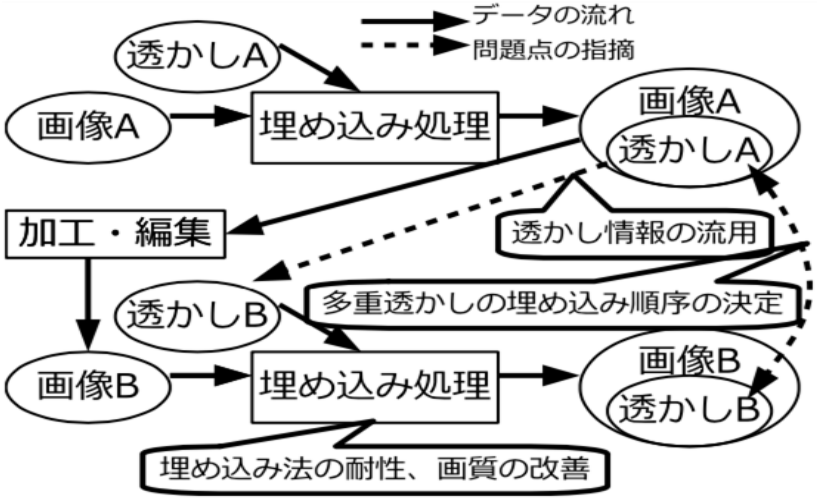
研究背景



・本研究では安全性と利便性を兼ね備えた次世代のデジタルコンテンツ著作権管理のあり方を探ることを目標とする。

・ 画像、動画、音声、自然言語など冗長性のあるメデイアに対して、証拠を残す手段としては電子透かしがある。

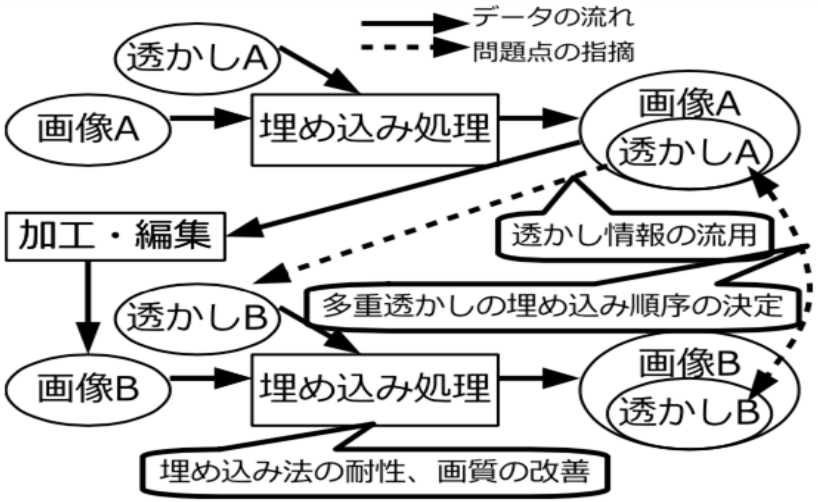
従来の電子透かしの問題



・まず一般的な電子透かしでは透かし情報がコピーされて他のコンテンツに流用される可能性がある。

・また、二次利用により多重に埋め込まれた電子透かしの優先順位や加工内容との対応関係を明確にできない。

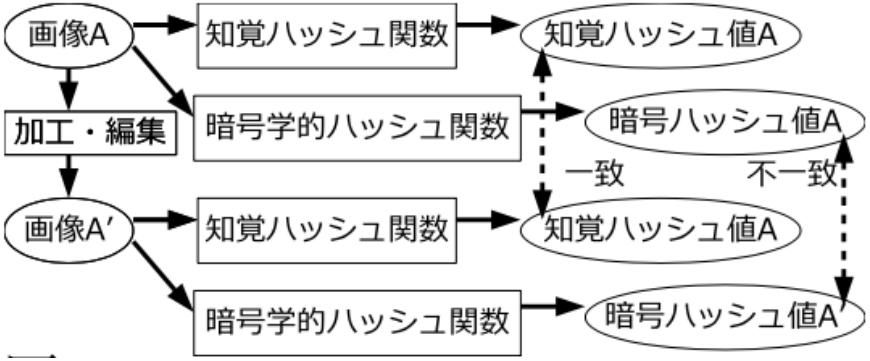
電子透かしの要件



・透かし情報の流用を防ぐには、原画像に依存した透かし情報を作る必要がある。

・不正な二次利用を防ぐために、著作権情報の提示だけでなく著作者の署名機能を保持するには、原画像に依存した透かし情報を作る必要がある。

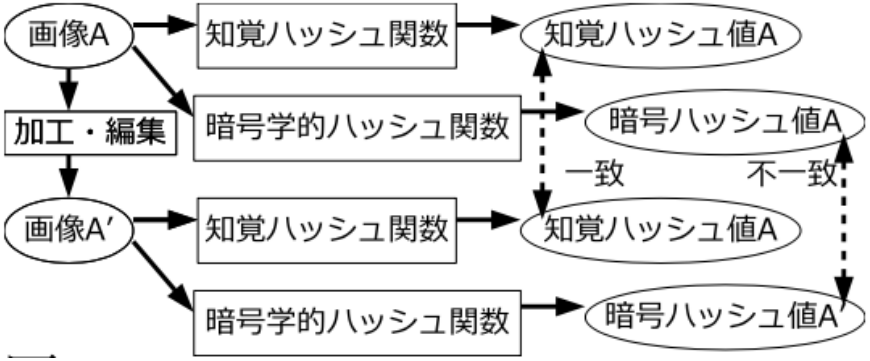
画像のメッセージダイジェストに基づく透かし情報の生成



・暗号学的ハッシュ関数を用いたメッセージダイジェスト

ビットごとのメッセージダイジェストなので加工編集などにより見た目が同じであってもハッシュ値が大幅に変化するため加工編集には対応できない。

画像のメッセージダイジェストに基づく透かし情報の生成

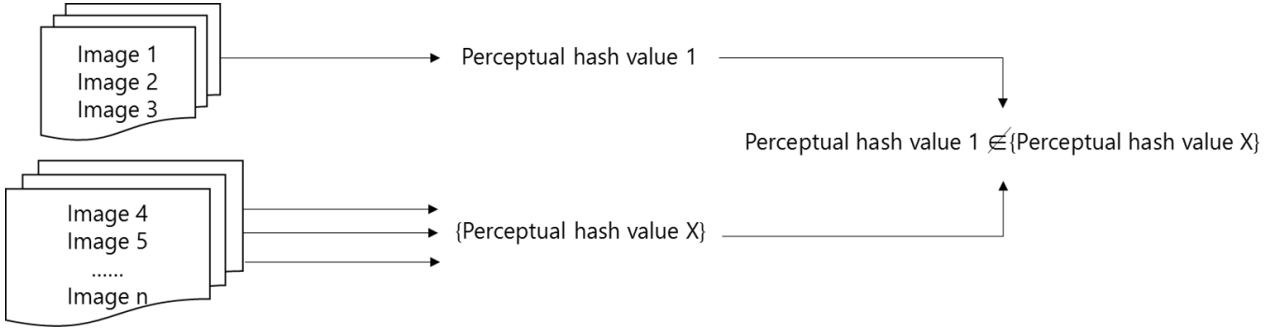


- 知覚ハッシュによるメッセージダイジェスト

  知覚ハッシュは画像などのメッセージダイジェストを人間の知覚に即した形で生成する手法である。

  加工、編集、非可逆符号化などを行っても原画像と同様のハッシュ値を得られる。
- 電子透かしの透かし情報を原画像の知覚ハッシュ値に基づいて生成する手法を提案する。

知覚ハッシュの要件



- 画像1、画像2、画像3をある画像とこれを加工編集した画像のグループとする、これらは知覚的に等しい画像なので、ほぼ同一（ハミング距離が閾値以下など）の知覚ハッシュ値１となる。
- 画像4、画像5から画像nを画像１、２、３、とは知覚的に異なる画像とするとき、生成される知覚ハッシュ値Xは知覚ハッシュ値１と異なるものとなる。

知覚ハッシュの種類

- 知覚ハッシュアルゴリズムには以下のようなものがある。

aHash	pHash	dHash	wHash
画像の平均輝度からの差分を使って算出	画像を離散コサイン変換(DCT)し周波数領域に変換後、低周波領域に対してaHashと同様の手法で算出	隣接領域との差分を使って算出	pHashのDCTの代わりに離散ウェーブレット変換(DWT)を用いて算出

従来の知覚ハッシュアルゴリズムの問題点

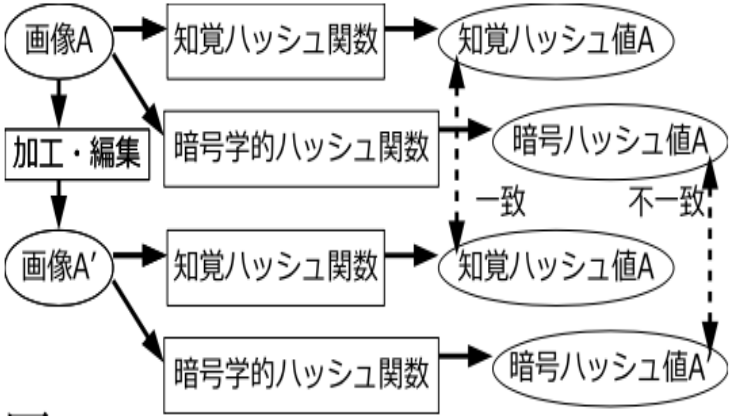
- 二次利用でよく用いられる画像の回転や反転といった加工・編集に対してハッシュ値が大きく異なってしまう。
- 非可逆符号化や拡大、ノイズ、などに対して最適なアルゴリズムが異なる。
- したがって、従来の知覚ハッシュアルゴリズムは、知覚ハッシュの要件を満たすことができない。

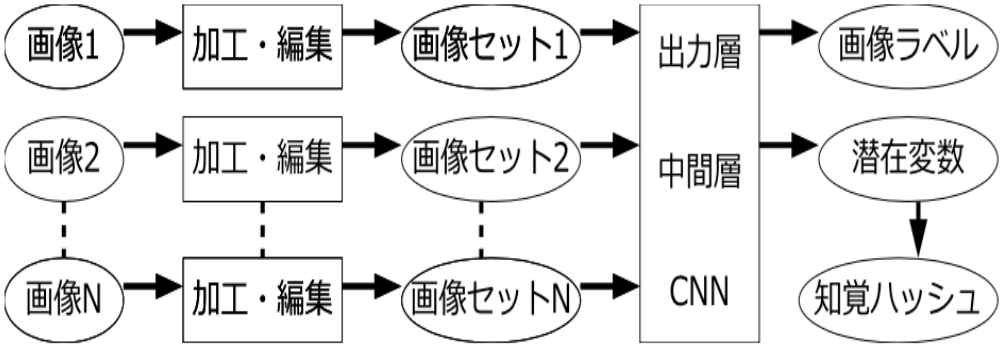
知覚ハッシュのハミング距離

	画像	知覚ハッシュ値（アベレージハッシュ）	ハミング距離
原画像		b69cbd89090b8f8e	0
回転画像		c0e2e091d2dcdfcb	35

- 左の表はアベレージハッシュというアルゴリズムにより計算したもの
- 原画像と同じ値ならハミング距離は0になる
- 従来の知覚ハッシュでは回転させただけでも値が異なる

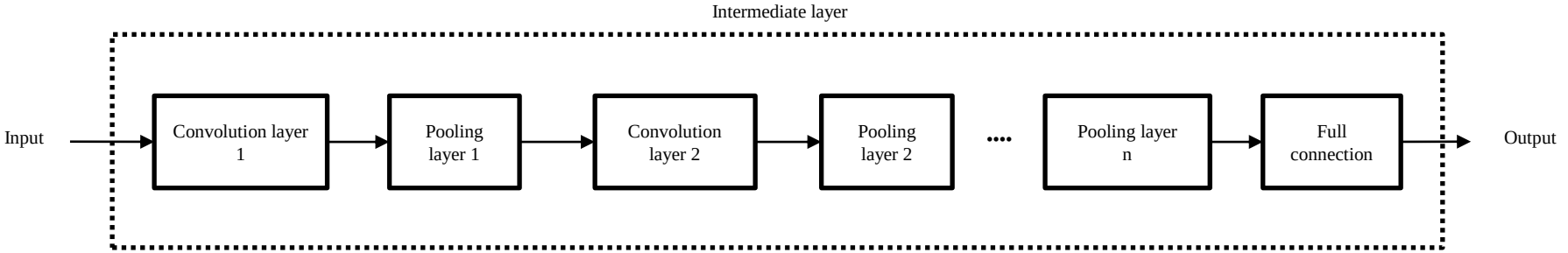
機械学習を利用した知覚ハッシュ





- そこで、深層学習の中間層の状態に基づいた新しい知覚ハッシュ関数の構成法を提案する。これにより既存の知覚ハッシュ関数と比較して加工編集に強い耐性が得られる。

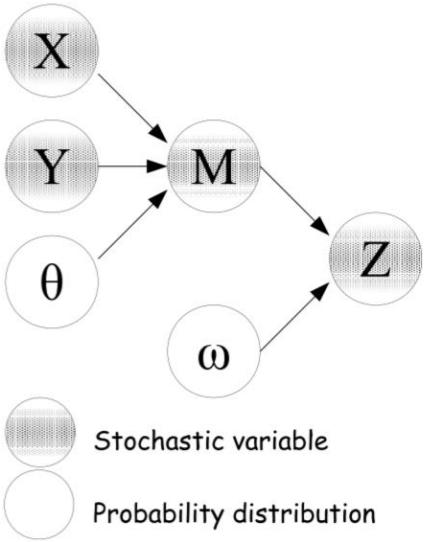
畳み込みニューラルネットワーク(CNN)



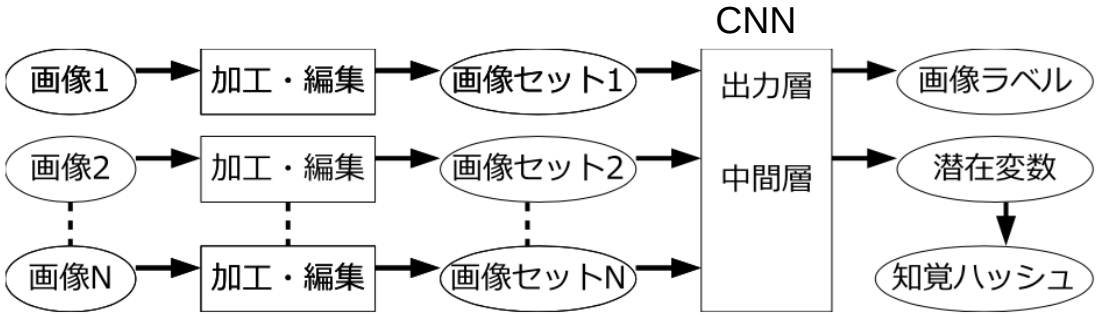
- 先ほどの知覚ハッシュの欠点を補うために,CNNを使用して画像の特徴量を抽出する。
- CNNの畳み込み層は、入力画像のさまざまな特徴を抽出する。プーリング層では、特徴として重要な情報を残しながら元の画像を縮小する。
- CNNには、ある程度の変換、スケーリング、回転不変性があると予想される。

CNNの中間層を用いた知覚ハッシュのグラフィカルモデル

- パラメータXは元の画像を表す。
- Yはさまざまな加工・編集後のXから形成された画像セットを表す。
- θは各加工・編集スタイルの確率変数である画像の加工・編集のためのパラメータを表す。
- これらの確率変数は、CNNへの入力として使用される。
- 確率変数Mは、 X 、 Y、およびθがCNNに入力された後の中間層出力を表す。
- 確率変数Mは画像および加工・編集後に変更されなかった知覚ハッシュ値を計算するために使用され、ωは潜在的確率変数を表し、 Mからハッシュ値Zを生成する。

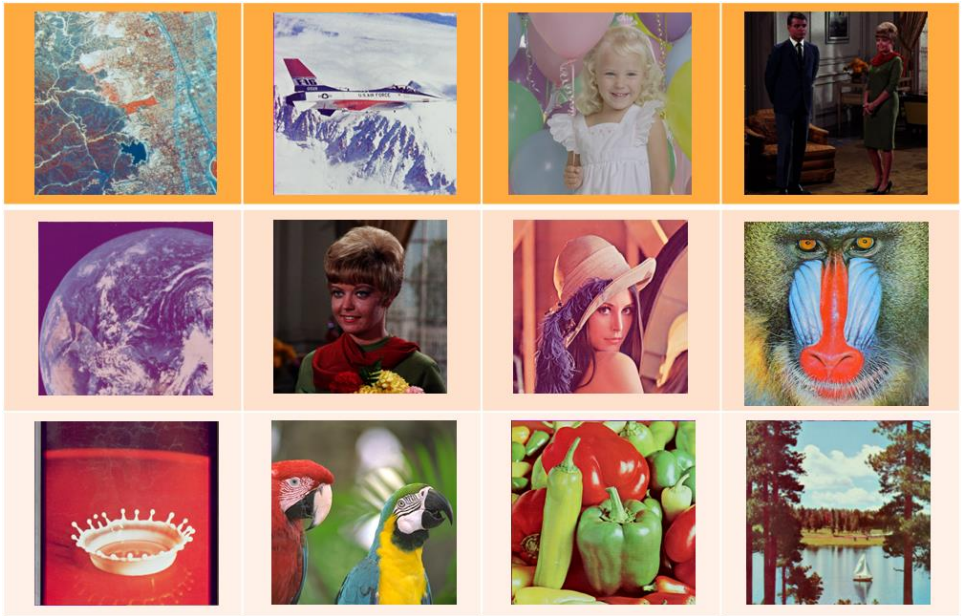


実験手順



- 図のように、画像データ（画像N）それぞれに対して加工・編集を施し画像セット(画像セットN)を用意する。
- 用意した画像セットに対して、CNNを学習させる。
- N個の画像セットの分類ができるCNNにおいて画像セット１内の画像から得られる中間層のデータは、同じ特徴を示すものである。
- つまり加工・編集に対しても強い耐性を持つデータであり、データを分析すれば原画像を示す新たな知覚ハッシュとなる。

データセット



- データセットとして画像処理の分野で広く利用されている12枚のカラー画像を使用する。

加工編集の種類

- ここで行う編集は、二次利用を想定した非可逆圧縮、ノイズ付加、アフィン変換である。
- 画像1枚につき3000枚の画像セットを生成する。
- このように学習データを増やす作業をデータオーグメンテーションという。

非可逆圧縮	JPEG方式を採用
ノイズ	画像にランダムノイズを付加
回転角度	左右に最大180度
スライド	左右上下に最大4ピクセル
反転	左右と上下に対して

使用するCNNのモデルと各層におけるパラメータの数

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
predictions (Dense)	(None, 1000)	4097000

Total params: 138,357,544
Trainable params: 138,357,544
Non-trainable params: 0

- CNNの実装には、Pythonの機械学習ライブラリのKerasで用意されている学習済みモデルのVGG16を使用する。
- このモデルは畳み込み13層＋全結合層3層＝16層のニューラルネットワークで構成されている。

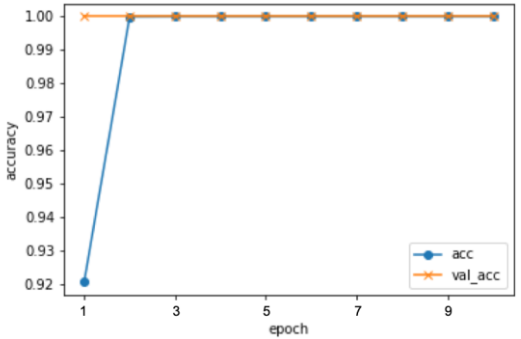
転移学習の利用

- 転移学習とは、ある設定（例えば分布P_1）で学んだことを別の設定（例えば分布P_2）の汎化能力向上に役立てようとする状況を指す言葉である。
- 転移学習のメリットはわざわざ1から自分で作成することなくCNNを利用できる。（CNNの設計には層の深さや学習率の設定といった調整が多い）
- すでに画像の分類に特化したモデルを利用できる。（VGG16はImageNetという100万枚以上の大規模画像データセットで学習済みのモデルであり、出力は1000クラスからなる画像の分類に特化したモデルであるため）
- 少ないデータセットでも高い分類精度を実現できる。（すでに大量のデータで学習済みなため、追加のデータは少量で済む）

転移学習の結果

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 150, 150, 3)	0
block1_conv1 (Conv2D)	(None, 150, 150, 64)	1792
block1_conv2 (Conv2D)	(None, 150, 150, 64)	36928
block1_pool (MaxPooling2D)	(None, 75, 75, 64)	0
block2_conv1 (Conv2D)	(None, 75, 75, 128)	73856
block2_conv2 (Conv2D)	(None, 75, 75, 128)	147584
block2_pool (MaxPooling2D)	(None, 37, 37, 128)	0
block3_conv1 (Conv2D)	(None, 37, 37, 256)	295168
block3_conv2 (Conv2D)	(None, 37, 37, 256)	590080
block3_conv3 (Conv2D)	(None, 37, 37, 256)	590080
block3_pool (MaxPooling2D)	(None, 18, 18, 256)	0
block4_conv1 (Conv2D)	(None, 18, 18, 512)	1180160
block4_conv2 (Conv2D)	(None, 18, 18, 512)	2359808
block4_conv3 (Conv2D)	(None, 18, 18, 512)	2359808
block4_pool (MaxPooling2D)	(None, 9, 9, 512)	0
block5_conv1 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv2 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv3 (Conv2D)	(None, 9, 9, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0

Total params: 14,714,688
Trainable params: 7,079,424
Non-trainable params: 7,635,264



- ここでは上図に示す全ての畳み込み層とプーリング層の重みを可変にすることで、より自前のデータセットに特化したモデルを作れる。
- 右図に示すように分類精度は100%となった。

4つの分析手法

方式１：プーリング層のノードの画素を次元とする主成分分析
方式２：プーリング層の16画素毎の平均値を利用した知覚ハッシュ
方式３：512次元のノードについての主成分分析
方式４：CNNの係数に基づく知覚ハッシュ

- 方式１から３は主成分分析を16次元についてと512次元についての各パターンについて行った。
- 方式４は転移学習を行った時の画像の特徴量がCNNの重みの状態変化に表れると考えた時の方式。

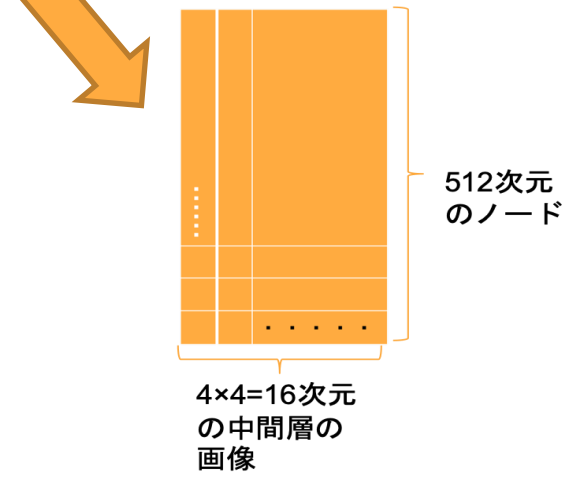
方式１

(a)画像セットから1000枚を準備。
(b)画像1枚毎に16次元について主成分分析。画像1枚につき512個の主成分が得られる。
(c)画像1000枚分の主成分の分散を、512次元のノード毎に求める。
(d)(c)で求めた分散と、分散の小さいノードとのペアがハッシュ値の基となる。

(a)画像の準備

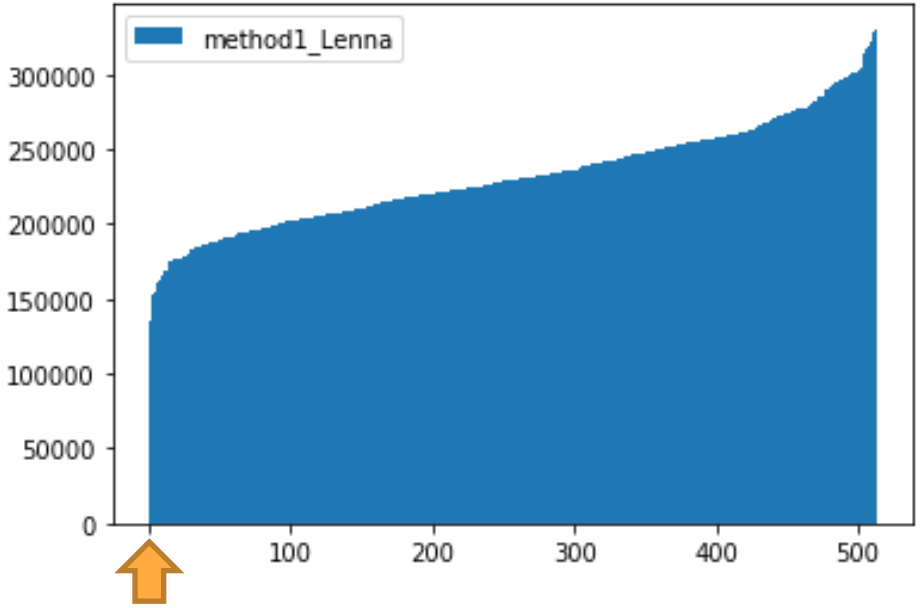
(b)画像1枚毎に主成分分析

(c)各ノード毎に1000枚分の分散を求める



Lennaにおける方式１

- Lennaの画像セットに対して方式1を行ったのが右図である。
- 横軸は512次元の各ノードであり、縦軸はノード毎に求めた画像1000枚分の主成分の分散である。
- 分散の小さいノードから抽出される画像の特徴量が、加工編集に影響されない画像の普遍的な特徴である。
- 得られた分散と分散の小さいノードの集合が、ハッシュ値を生成する基となる。



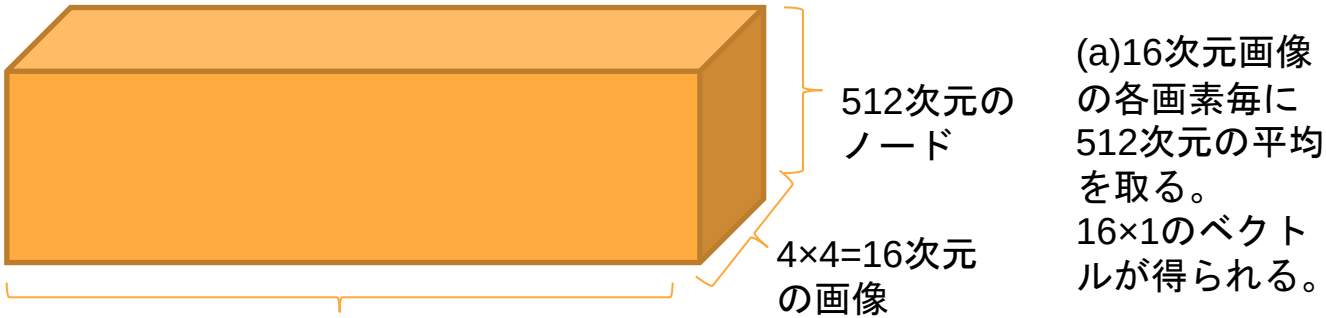
Lennaでは10番目のノードの分散が最も小さくなった

方式2

(a)4×4=16次元の各次元について512次元のノードの平均値をとる。16×1のベクトルが得られる。

(b)Lennaの原画像(origin)のベクトルをハッシュ値とし、画像セット(all)、左右反転(horizontal)、上下反転(vertical)、非可逆圧縮(jpg)、ノイズ付加(noise)のそれぞれの加工画像についてもベクトルを求める。

(c)異なる原画像についても(a)(b)と同様にベクトルを求める。



方式２の比較（Lennaの原画像を基準）

- (d)ハッシュ値を基準に(b)(c)で求めたそれぞれのベクトルについて相関係数を求める。これらを示したのが下と右の表。

比較対象	ピアソン相関係数
画像セット内の画像（1000枚）	83.9%
非可逆圧縮	99.9%
ノイズ	98.2%
左右反転	88.9%
上下反転	51.4%

比較対象	ピアソン相関係数
Aerial	66.7%
Airplane	46.5%
Balloon	67.0%
Earth	59.4%
Girl	73.9%
Mandrill	55.8%
Parrots	70.8%
Pepper	77.3%
Sailboat	77.1%
couple	47.3%
milkdrop	70.7%

方式２の閾値

- (e)相関係数の結果から、Lennaの画像セットとLenna意外の原画像については相関係数80%でほとんど分けられる。
- (f)閾値として相関係数を80%と設定することで、Lennaの原画像とLennaのほとんどの加工画像の相関係数が80%以上になり、同じ画像セットと判定できる。

方式３（ハッシュ値の生成）

(a) 512次元について主成分分析を行う。画像1枚につき16×1のベクトルが得られる。
(b)各加工画像と原画像について(a)を行う。
(c)目的の原画像について得られたベクトルを基準に、各加工画像と、異なる原画像とのベクトルとの相関係数を求める。

(a)512次元の主成分を求める

方式３（閾値の設定）

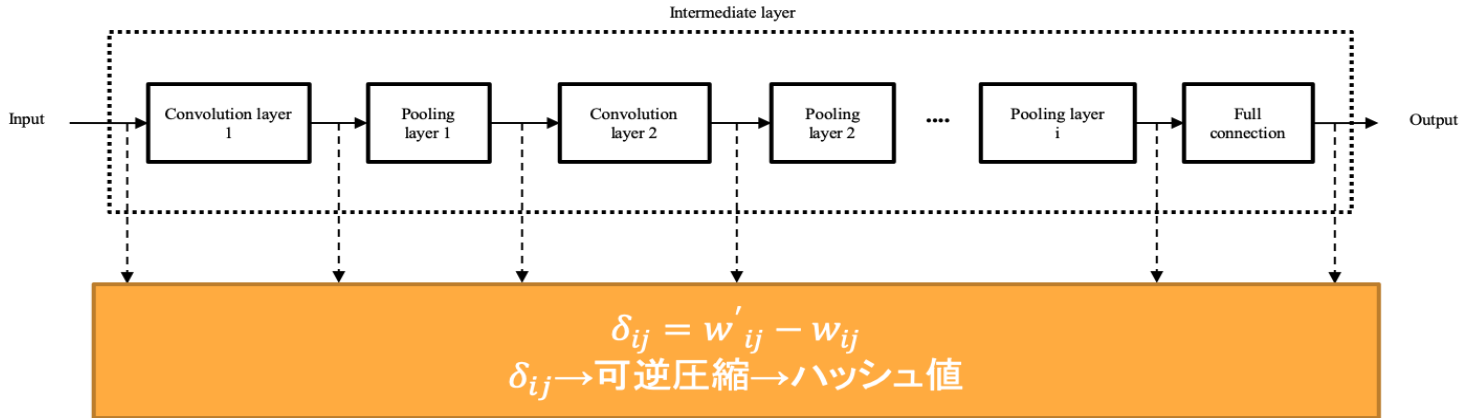
(d)相関係数を基に、同じ画像セット内の画像と異なる原画像との境目となる閾値を求める。
(e)目的の原画像について得られたベクトルをハッシュ値とし、(d)で求めた閾値の範囲内の画像を同じ画像セットと判定する。

方式４（知覚ハッシュの生成）

- 目的とする原画像を受理するか受理しないかの2種類のクラスを出力とするCNN を考える。
- 1. 知覚ハッシュの生成
- (a) 方式 1,2 と同様に原画像および編集加工した36000枚ほどの画像を用いてVGG16 において学習させる。
- (b) 学習の対象となったi層のノードjについて、学習前の重み係数を w_{ij}、学習後の重み係数をw'_{ij}、とする。また、学習前後の重み係数の差分を$\delta_{ij} = w'_{ij} - w_{ij}$とする。
- (c) $i \times j$個の、係数の差分のδ_{ij}を連結したものをLempel-Ziv アルゴリズムとハフマン符号により可逆圧縮したものを知覚ハッシュ値とする。

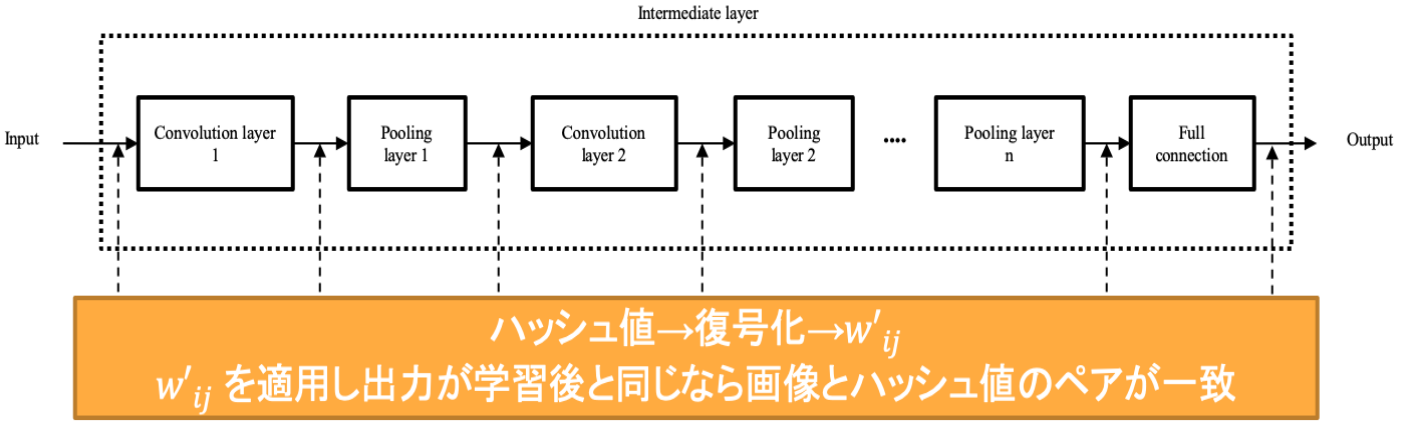
方式４におけるハッシュ値の生成の図

- (a) 方式 1,2 と同様に原画像および編集加工した36000枚ほどの画像を用いて VGG16 において学習させる。
- (b) 学習の対象となったi層のノードjについて、学習前の重み係数を w_{ij}、学習後の重み係数をw'_{ij}、とする。また、学習前後の重み係数の差分を$\delta_{ij} = w'_{ij} - w_{ij}$とする。
- (c) $i \times j$個の、係数の差分のδ_{ij}を連結したものをLempel-Ziv アルゴリズムとハフマン符号により可逆圧縮したものを知覚ハッシュ値とする。



方式４（知覚ハッシュの検査）

- 2. 知覚ハッシュの検査（下図）
- (a) 圧縮された係数の差分情報を、知覚ハッシュ値を復号化して取り出す。
- (b) CNN に係数の差分情報を適用して学習済みの状態を再現する。
- (c) 画像を入力し、受理状態にクラス分けされれば正当な画像と知覚ハッシュのペアであると判定する。



まとめ

- 方式１と方式２について実験を行った
- 方式1では、中間層の各ノードにおいて、画像セット毎に、加工に影響を受けないノードを求めた。そのため、得られたノードとノードの分散は、画像の普遍的な特徴を示す情報が含まれており、これを用いてハッシュ値が生成可能となる 。
- 方式２では、中間層から得られた画像の画素毎に平均値を計算し、16×1のベクトルを求めることでこれをハッシュ値とした。加工画像と、異なる原画像との相関係数を比較することで閾値を求めたところ、ほとんどの場合において画像の分類が可能となった。

今後の課題

- 本実験で一部の加工画像については、ハッシュ値の精度が低かったため、より画像の特徴の抽出の精度を上げる必要がある。
- 本実験では12種類の画像セットの分類であったが、さらに画像の種類が増えた場合、どのように対応するか。例えば、画像セットの分類を、目的の画像セットと、それ以外の画像群との2分類にするという対応方法がある。
- 方式３と４では実装には至らなかったため、実装することで、どのような結果が得られるか比較検討の必要がある。

発表論文

1) 中谷　憲・森住哲也・木下宏揚　“ベイジアンモデルによる情報漏えい分析のための機械学習”2018年電子情報通信学会ソサイエティ大会A-12.技術と社会・倫理
2018年9月12日発表
2) 中谷　憲・孟　昭雄・森住哲也・木下宏揚 “畳み込みニューラルネットワークを用いた知覚ハッシュのための中間層の分析”2019年電子情報通信学会SITE インターネットと情報倫理, 一般
2019年12月6日発表

使用する中間層

- 本実験では５番目のプーリング層を使用する。この層では4×4=16次元の画像が512個(512次元)出力される。
- 中間層データは最も出力に近いプーリング層を使用する。
- 入力層に近い層では各ノードのヒートマップが原画像に近い状態なので構造情報など特徴的な要素の解析には不向きである。
- また、畳み込み層で各フィルタがそれぞれバラバラに抽出した画像の特徴量が、プーリング層でまとめられ、入力画像の特徴が単純化されたプリミティブな情報に分解されているため、図に示すように、分析時に次元圧縮を行っても、中間層のデータの寄与率が大きいためでもある。