

# データ分析コンテストの 勝者解答から学ぶ

---



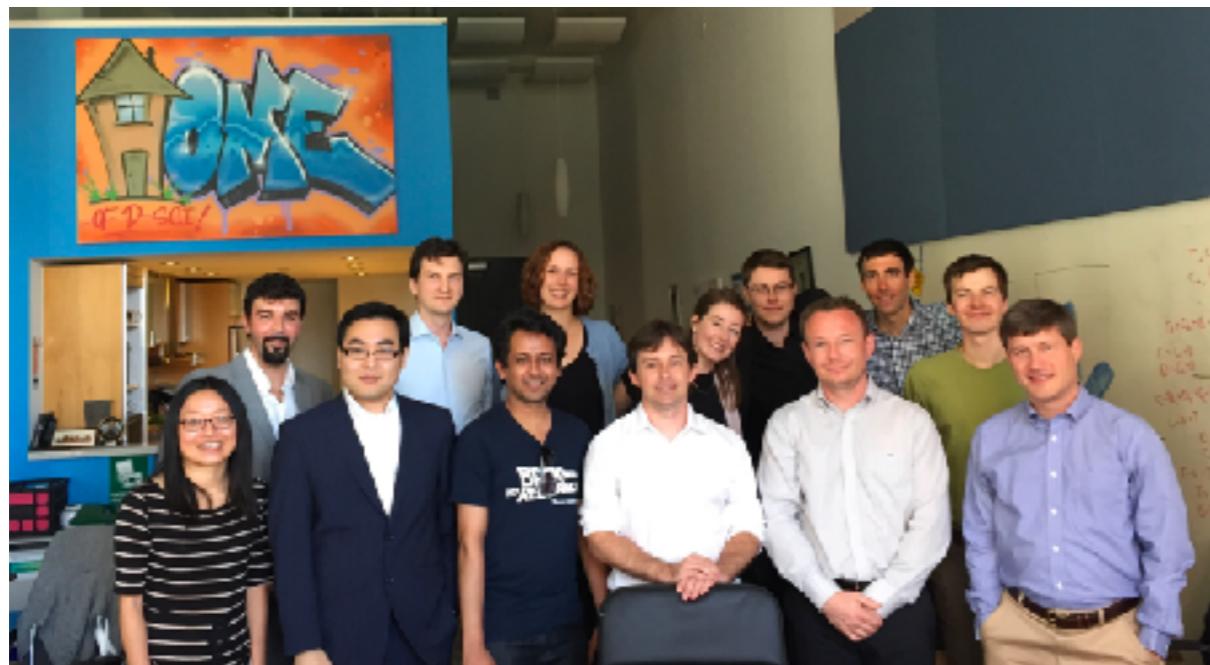
Kohei Ozaki (@smly)  
Recruit Technologies  
Advanced Technology Lab (ATL)

# 自己紹介

リクルートテクノロジーズ ATL の Sr. Software Engineer です。

- Kaggle 歴 7 年 (Grandmaster, Highest rank: 4th)
- Kaggle (Top10 finishes x13, Prize x3) ※前回の講演から 3 個増えました
- TopCoder Marathon Match (3 WINS)
- ACM/KDD, KDD Cup 2015 1st prize winner

▼ Kaggle オフィスでの集合写真 (サンフランシスコ)



▼ リクルートテクノロジーズ ATL (広尾)



# 今日の話

- ここ数年のKaggleとデータ分析コンテスト (10) 一般
- どのようにしてデータ分析コンテストで勝つか (10) 一般
- 最近のコンテストにおける解法の解説 (40) 応用  
ここが本題

今回は一般論よりやや各論に踏み込んだ話をしてみます。

# 今日の話

- ここ数年のKaggleとデータ分析コンテスト (10)
- どのようにしてデータ分析コンテストで勝つか (10)
- 最近のコンテストにおける解法の解説 (40)

今回は一般論よりやや各論に踏み込んだ話をしてみます。

# 振り返り：NN萌芽・ツール開発加熱 (12-14)



- 2012年ごろは Steffen Rendle の libFM (Factorization Machines [2]) や Rie Johnson の RGF (Regularized Greedy Forest [3]) など、アルゴリズムの新規開発によって他の差をつけて勝利する場面が何度かあった。
- 機械学習やデータマイニングを活用する人工が増加、新しい道具が次々に生まれた。Lasagne, cxxnet, keras, XGBoost など。開発者本人がコンテストを通じてソフトウェアのベンチマークを行ったり宣伝するなどしていた。

# 振り返り：NN萌芽・ツール開発加熱（12-14）

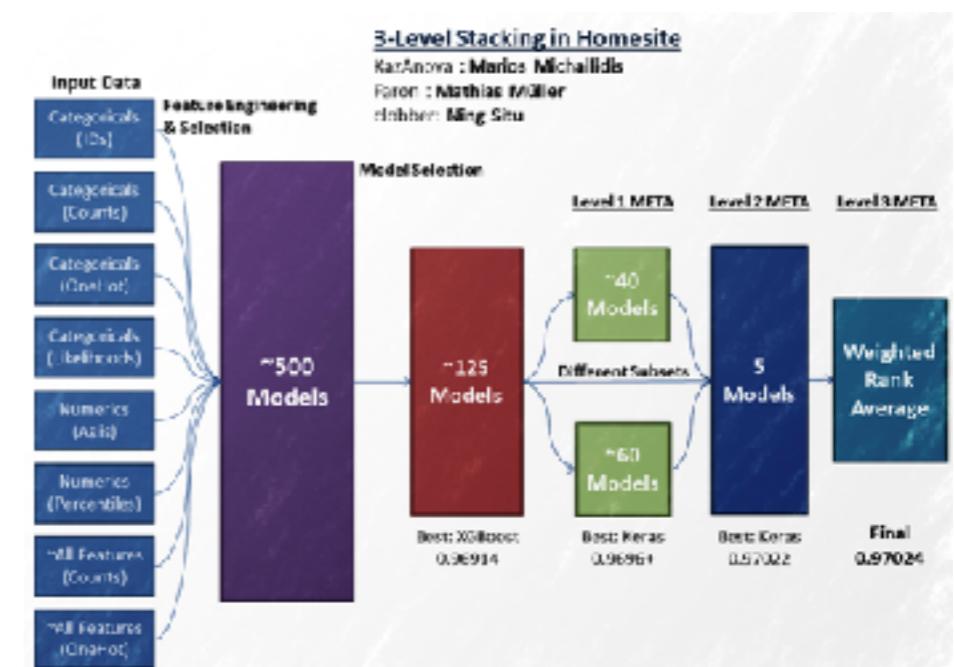


- 2012年の ImageNet (ILSVRC) におけるブレークスルーは大きな衝撃ではあったが、まだ GPU を扱う環境を整備したり CUDA を使ったプログラムを Kaggle で使うユーザーは少数派であった。
- 2013年には Job Salary Prediction Challenge で Hinton の研究室の OB がスパース特徴量を入力とした MLP を実装して他を圧倒。

# 振り返り：フランケンシュタイン (14-16)



- 大量にモデルを作りスタッキングするという手法が流行る。与えられた指標で少しでも上回っていれば勝利するというコンテストの仕組み上、トップ争いが僅差であればモデルを増やすことにインセンティブが生まれる。
- 500モデル、1000モデルとアンサンブルに使うモデル数に際限がなくなり、frankenstein ensemble などと揶揄されることも。



# 振り返り：フランケンシュタイン (14-16)



一応フォローすると、Kaggle はコンテスト入賞者にソリューション内の知見をドキュメントとして用意させる（どのような特徴量を使い、気付いたかなど）。

アンサンブルで差がつくコンテストは、そもそもタスク設計が単純すぎるとか、工夫の余地がないとか、他と差をつけることが難しい。コンテストに向いていないことが根本的な問題であると思う。

# 振り返り：NN時代・大規模データ (16-)



- GPU が研究者や開発者などに広く普及し、DL 関連の研究開発とその成果物のリリースサイクルが高速になり、多くのタスクで応用可能であると認識されるようになる。競技者の解法もバリエーションが豊かになった。
- データセットも大規模なマルティメディアデータ（信号、音声、画像、動画）が増え、3TB のコンテストデータをダウンロードする場面もできた。

# 振り返り：Kernel・強化学習コンペ (17-)



- 計算資源と精度のトレードオフが注目されはじめる。Two Sigma, Mercari などの計算資源や実行環境に制約を加えたコンテストが登場。ほか TopCoder 開催も SpaceNet Challenge も計算資源の制約を加えている
- 2018 年には強化学習などのコンテストもプランにあると表明され、ユニークなタスクのコンテストが増えていく\* と思われる。

\*出典："Reviewing 2017 and Previewing 2018" blog.kaggle.com/2018/01/22/reviewing-2017-and-previewing-2018/

# 言いたかったことを整理すると



- ・ アンサンブルの競い合いは不毛だが、問題設定にも原因がある。
- ・ NN の研究開発が活発になったおかげで解法もバリエーションが豊富になった。
- ・ 計算資源の制約を入れたり、強化学習を題材にするなど変化を続けている。

# 今日の話

- ここ数年のKaggleとデータ分析コンテスト (10)
- どのようにしてデータ分析コンテストで勝つか (10)
- 最近のコンテストにおける解法の解説 (40)

今回は一般論よりやや各論に踏み込んだ話をしてみます。

# どうやつたら勝てるのか

大雑把に考えると今も昔も必要なことは変わらない。

EDA (探索的データ分析)

Validation (適切なモデル評価)

Survey (研究成果や過去解答から学ぶ)

# 探索的データ分析 (EDA)

様々な切り口でデータを見て、可視化して、仮説を立て、問題の解き方を考える。とても重要なフェーズ。

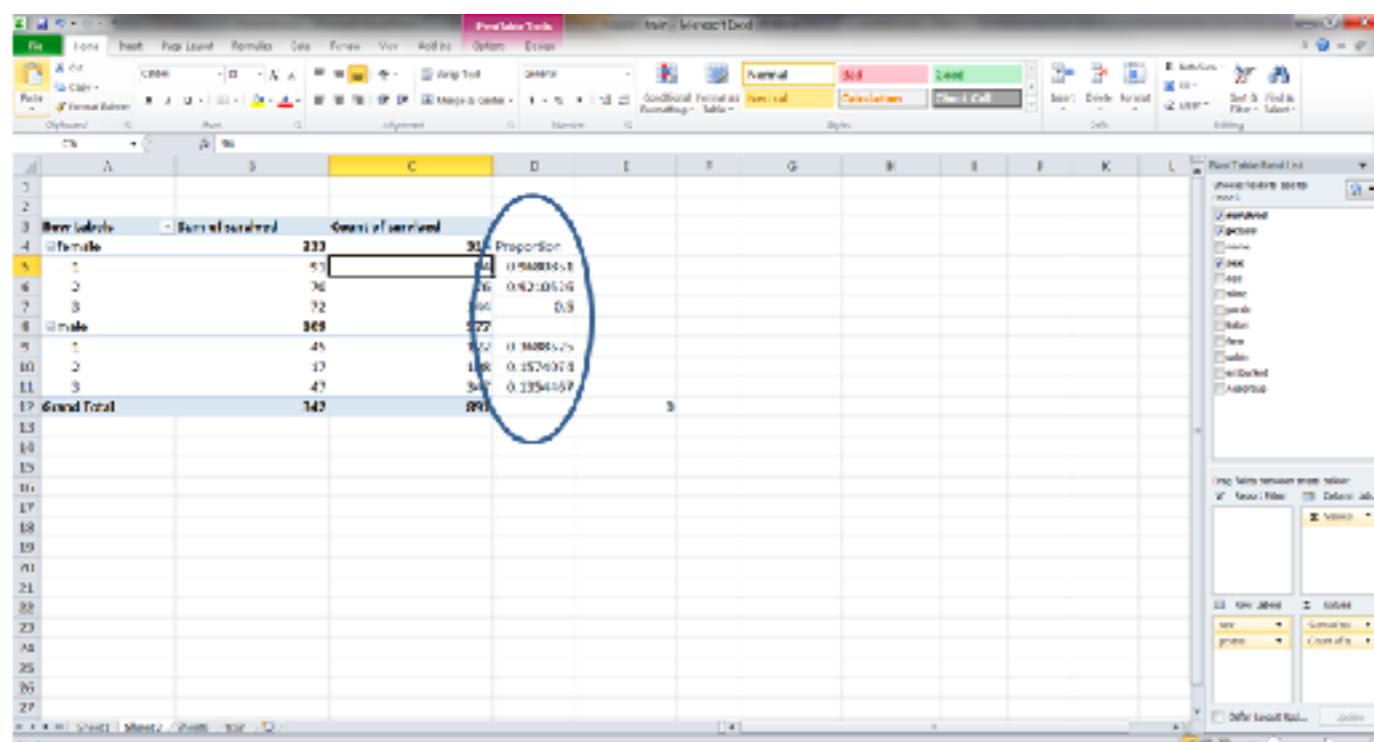
期待する成果：

- ・ドメイン知識やデータのパターンからモデル・解法を設計する
- ・タスク背景からデータの規則性を推測して特徴量を作るなど

集計（集約）と可視化を道具として、アイディアを産む

# 集計・集約処理、可視化によるデータ理解

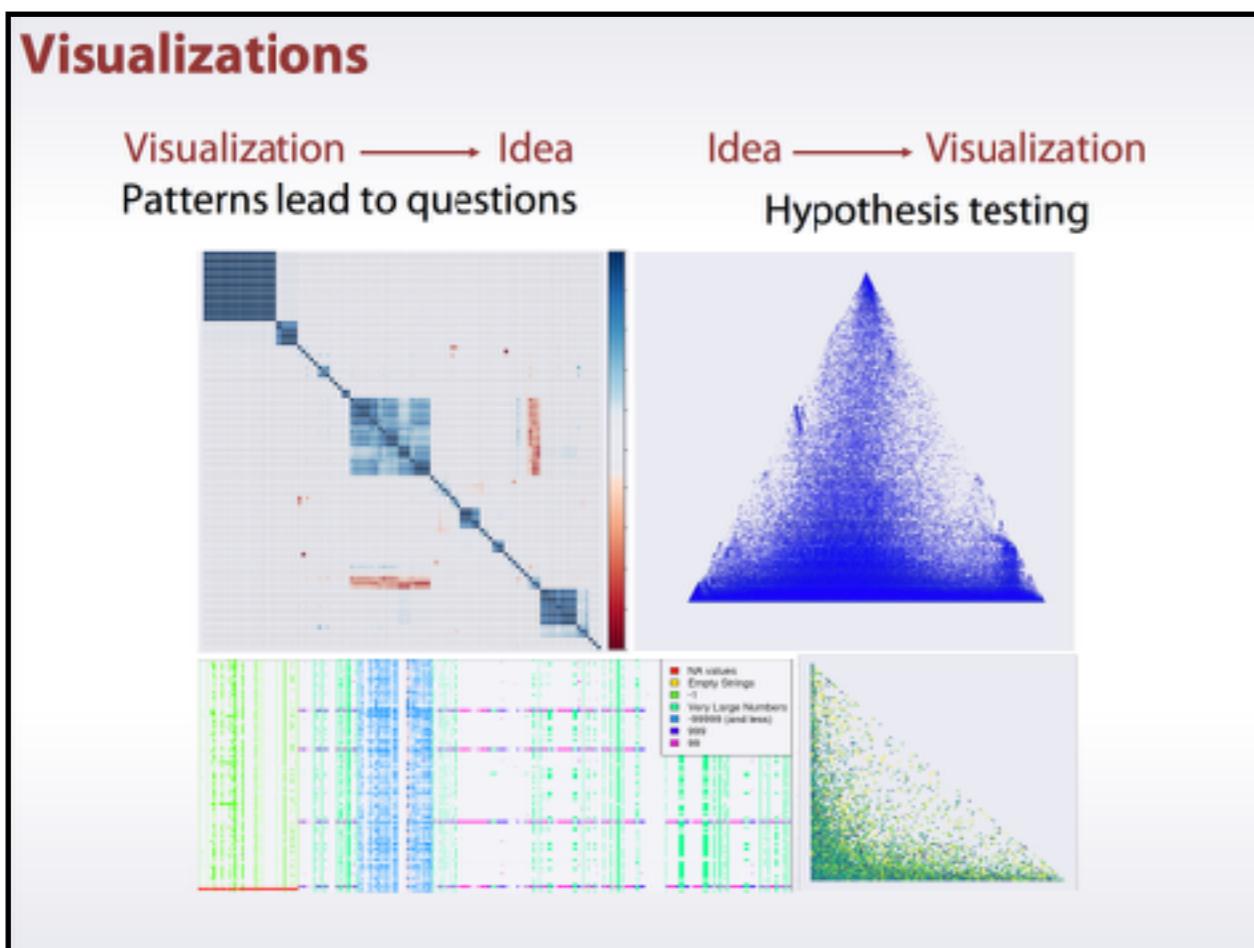
- ・ 小規模な構造化データであれば Excel のピボットテーブルも有用
- ・ 画像データであればパターンごとにグループ化して並べたり、  
予測モデルのロスが大きい事例・小さい事例を俯瞰してみる
- ・ 必要に応じて可視化ツールを自作する



出典：<https://www.kaggle.com/c/titanic/discussion/28323>

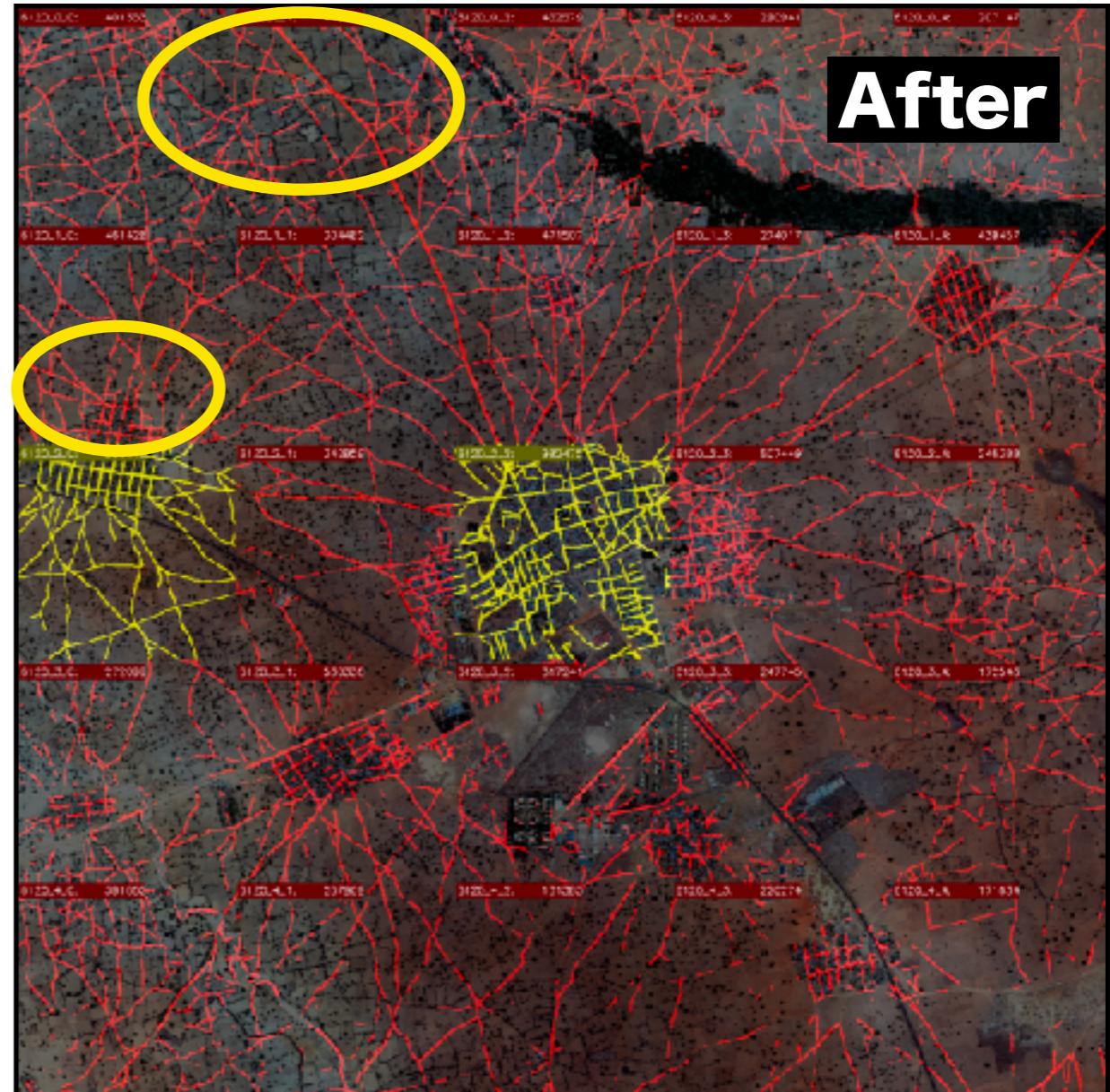
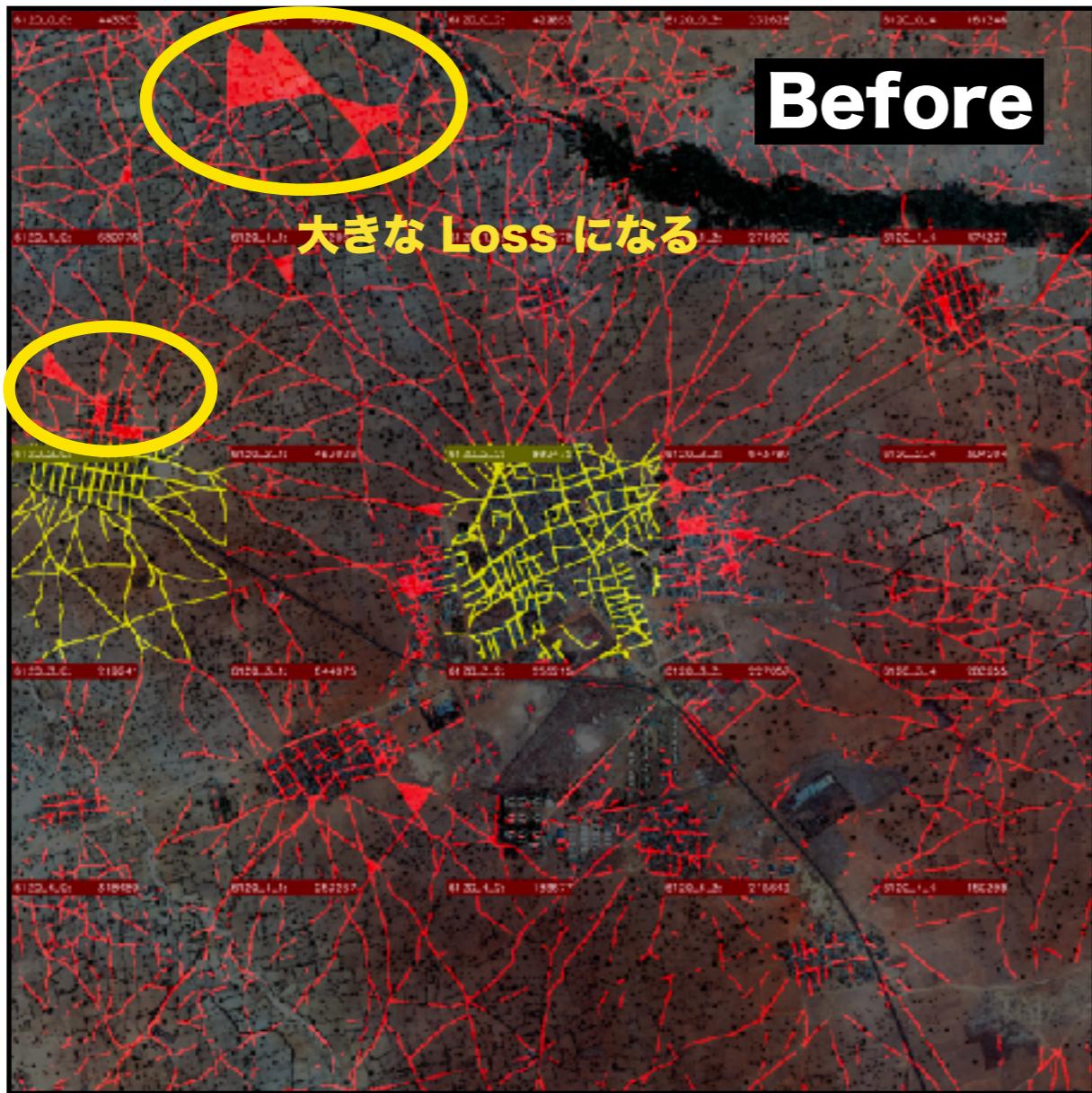
# データを可視化する：仮説検証とアイディア

- ・道具として可視化を活用する
- ・可視化→アイディア（質問につながるパターンを探す）
- ・アイディア→可視化（仮説検証）



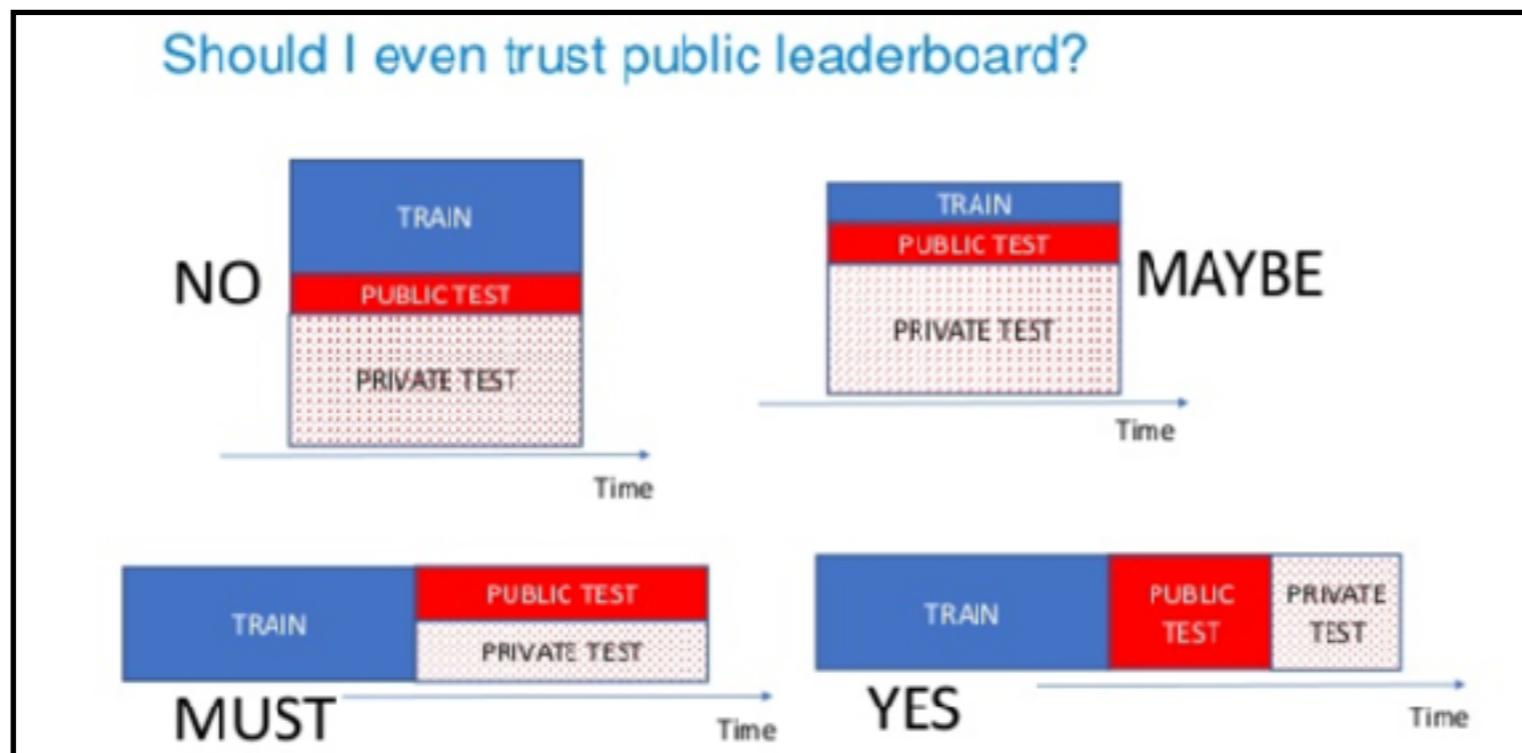
# データを可視化する：エラー分析

衛星画像の semantic segmentation で予測マスクのエンコーディングにバグがあることに気づくことが出来たケース。



# 適切なモデル評価 (Public LB を使うべきか)

- データセットの分割ごとに適切な Validation セットを用意する
- Public LB はテスト事例が少ないときは特に気をつける



出典: “Tips and tricks to win kaggle data science competitions”

“<https://www.slideshare.net/DariusBaruauskas/tips-and-tricks-to-win-kaggle-data-science-competitions>”



Kaggle のフレームワークにおける過学習を考えると、理論的には「分割の割合」ではなく「テスト事例数」に影響される問題であるはず (ref: [1])

# 過学習の直感的な例：Boosting Attack

単純化した  $N$  個のテスト事例の二値分類問題を考える：

$N$  次ベクトル  $y \in \{0, 1\}^N$  を当てる。評価指標はエラー率とする。

ある予測  $y_i \in \{0, 1\}^N$  の Public LB スコアを  $s_H(y_i)$  とする。

正解	$y =$	<table border="1"><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr><tr><td>0</td></tr><tr><td>1</td></tr></table>	1	0	0	1	0	0	0	1	Public						
1																	
0																	
0																	
1																	
0																	
0																	
0																	
1																	
		<table border="1"><tr><td>予測 1</td></tr><tr><td><math>y_1 =</math></td></tr><tr><td><table border="1"><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr></table></td></tr></table>	予測 1	$y_1 =$	<table border="1"><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr></table>	1	1	1	0	0	1	1	1	<table border="1"><tr><td>Public LB Score 1</td></tr><tr><td><math>s_H(y_1) = 0.75</math></td></tr></table>	Public LB Score 1	$s_H(y_1) = 0.75$	
予測 1																	
$y_1 =$																	
<table border="1"><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr></table>	1	1	1	0	0	1	1	1									
1																	
1																	
1																	
0																	
0																	
1																	
1																	
1																	
Public LB Score 1																	
$s_H(y_1) = 0.75$																	
		<table border="1"><tr><td>予測 2</td></tr><tr><td><math>y_2 =</math></td></tr><tr><td><table border="1"><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr></table></td></tr></table>	予測 2	$y_2 =$	<table border="1"><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr></table>	1	0	1	1	1	1	1	0	0	<table border="1"><tr><td>Public LB Score 2</td></tr><tr><td><math>s_H(y_2) = 0.25</math></td></tr></table>	Public LB Score 2	$s_H(y_2) = 0.25$
予測 2																	
$y_2 =$																	
<table border="1"><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>1</td></tr><tr><td>0</td></tr><tr><td>0</td></tr></table>	1	0	1	1	1	1	1	0	0								
1																	
0																	
1																	
1																	
1																	
1																	
1																	
0																	
0																	
Public LB Score 2																	
$s_H(y_2) = 0.25$																	

# Boosting Attack [Blum & Hardt '15]

Algorithm (Boosting Attack):

ランダムな予測から Public Score の良い結果を選択する

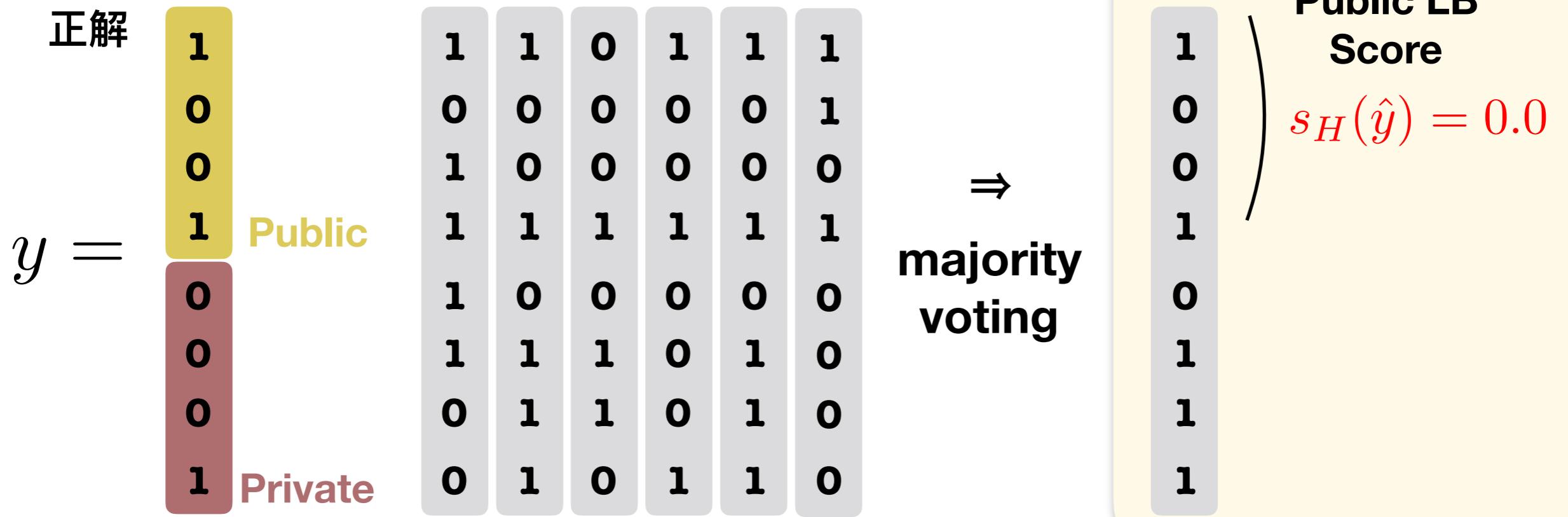
正解	1	0	0	0	1	0	1	1	0	0	0	1
$y =$	0	1	1	0	0	0	1	0	1	1	1	0
	0	0	0	0	1	0	1	0	1	0	0	1
	0	0	0	0	0	1	0	1	1	1	0	0
	0	0	0	0	0	1	1	1	1	1	0	0
	0	0	0	0	0	1	1	1	1	1	1	0
	0	0	0	0	0	1	1	1	1	1	1	0
	1	1	0	1	1	1	0	0	1	0	1	1

ランダムに予測のベクトルを作成する

# Boosting Attack [Blum & Hardt '15]

Algorithm (Boosting Attack):

ランダムな予測から Public Score の良い結果を選択する

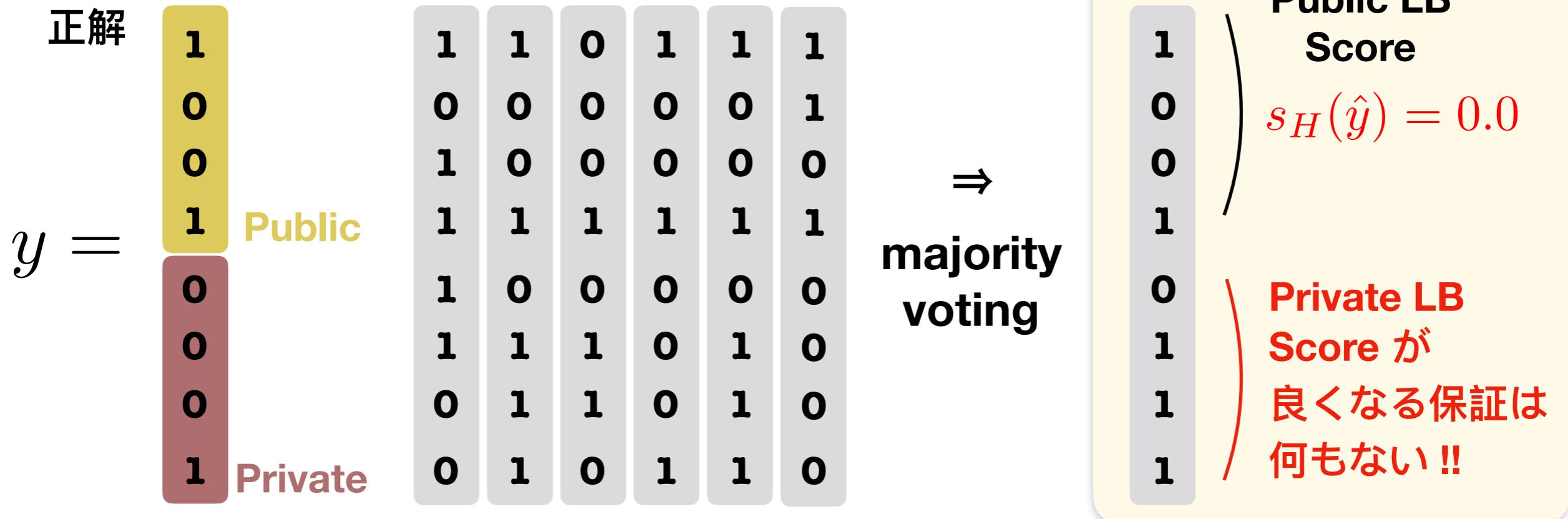


Public LB スコアの良いベクトル  
 $s_H(y_i) < 0.5$  だけを選ぶ

# Boosting Attack [Blum & Hardt '15]

Algorithm (Boosting Attack):

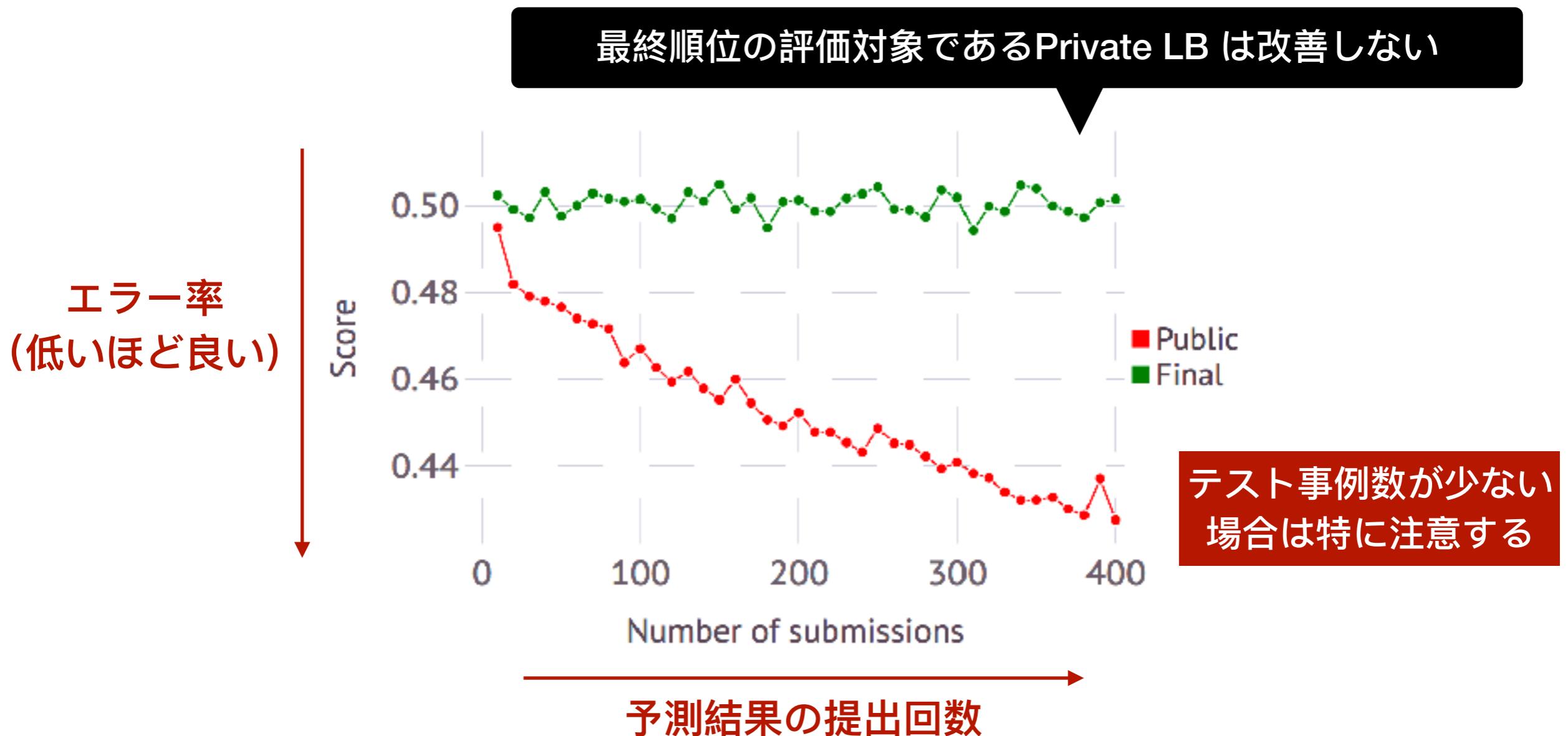
ランダムな予測から Public Score の良い結果を選択する



Public LB スコアの良いベクトル  
 $s_H(y_i) < 0.5$  だけを選ぶ

# Boosting Attack [Blum & Hardt '15]

最終的な順位を決定する Private LB スコアは改善しない。テスト事例数が多くれば、Public のエラー率の分散は小さくなるので乖離も小さくなる。



# サーベイ：分野独特のアプローチを学ぶ

## ■ 赤外分光法による土壤調査の観測データに対する前処理例 (Africa Soil Property Prediction Challenge) :

(1) Savitzky-Golay filter (2) Continuum Removal (3) Discrete wavelet transforms (4) First Derivatives (5) Unsupervised Feature Selection (6) Log transform …

<https://www.kaggle.com/c/afsis-soil-properties/discussion/10825>

## ■ 脳磁図記録の前処理・特徴量抽出 (DecMeg2014 Challenge) :

5-order Butterworth band-pass filter, Spatial Filtering, Riemannian Mean, Tangent Space Mapping …

<https://github.com/alexandrebarachant/DecMeg2014>

<https://web.archive.org/web/20160328051018/http://alexandre.barachant.org/wp-content/uploads/2014/08/documentation.pdf>

# サーベイ：深層学習の実践的な研究成果

---

使えそうなものはチェックする。ここ1～2年の成果が使われることも珍しくない。本スライドで紹介する上位解法が参考にした文献の例：

[Zhang+ '17] "mixup: Beyond Empirical Risk Minimization", <https://arxiv.org/abs/1710.09412>

[Miech+ '17] "Learnable pooling with Context Gating for video classification", <https://arxiv.org/abs/1706.06905>

[Tian+ '16] "Detecting Text in Natural Image with Connectionist Text Proposal Network", In Proc. of ECCV '16 <https://arxiv.org/abs/1609.03605>

[Qi+ '17] "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation", In Proc. of the CVPR '17 <https://arxiv.org/abs/1612.00593>

[Ma+ '17] "Multi-View Deep Learning for Consistent Semantic Mapping with RGB-D Cameras", In Proc. of the IROS '17 <https://arxiv.org/abs/1703.08866>

# 今日の話

- ここ数年のKaggleとデータ分析コンテスト (10)
- どのようにしてデータ分析コンテストで勝つか (10)
- 最近のコンテストにおける解法の解説 (40)

今回は一般論よりやや各論に踏み込んだ話をしてみます。

# 最近のコンテストにおける解法の解説

2つのコンテストにおける解法を紹介する。



Featured Prediction Competition

~ 2017/12/14

**Cdiscount's Image Classification Challenge**

Categorize e-commerce photos

Cdiscount · 627 teams · 2 months ago

\$35,000 Prize Money

This thumbnail shows a modern building with large glass windows and brick walls. A trophy icon and the text 'Featured Prediction Competition' are in the top left. An orange box in the top right contains the deadline '~ 2017/12/14'. The challenge title 'Cdiscount's Image Classification Challenge' is in bold at the top, followed by the description 'Categorize e-commerce photos'. Below that is the sponsor logo 'Cdiscount' and the participation stats '627 teams · 2 months ago'. On the right, it says '\$35,000 Prize Money'. The bottom right features the Statoil logo.



Featured Prediction Competition

~ 2018/01/23

**Statoil/C-CORE Iceberg Classifier Challenge**

Ship or iceberg, can you decide from space?

Statoil · 3,343 teams · a month ago

\$50,000 Prize Money

Data and Remote Sensing expertise provided by

C-CORE

This thumbnail features a large iceberg against a dark background. A trophy icon and the text 'Featured Prediction Competition' are in the top left. An orange box in the top right contains the deadline '~ 2018/01/23'. The challenge title 'Statoil/C-CORE Iceberg Classifier Challenge' is in bold at the top, followed by the description 'Ship or iceberg, can you decide from space?'. Below that is the sponsor logo 'Statoil' and the participation stats '3,343 teams · a month ago'. On the right, it says '\$50,000 Prize Money'. The bottom right features the C-CORE logo with the text 'Data and Remote Sensing expertise provided by'.

# 最近のコンテストにおける解法の解説

2つのコンテストにおける解法を紹介する。



Featured Prediction Competition

~ 2017/12/14

**Cdiscount's Image Classification Challenge**

Categorize e-commerce photos

Cdiscount · 627 teams · 2 months ago

\$35,000 Prize Money

This thumbnail for a Kaggle competition features a photograph of a modern building with large glass windows. A trophy icon and the text 'Featured Prediction Competition' are in the top left. An orange box in the top right contains the deadline '~ 2017/12/14'. The competition title 'Cdiscount's Image Classification Challenge' is in bold at the top, followed by the task description 'Categorize e-commerce photos'. The host 'Cdiscount' and participation stats ('627 teams · 2 months ago') are in the bottom left. The total prize money '\$35,000 Prize Money' is on the right.



Featured Prediction Competition

**Statoil/C-CORE Iceberg Classifier Challenge**

Ship or iceberg, can you decide from space?

Statoil · 3,343 teams · a month ago

\$50,000 Prize Money

Data and Remote Sensing expertise provided by C-CORE

This thumbnail for a Kaggle competition features a background image of icebergs in the ocean. A trophy icon and the text 'Featured Prediction Competition' are in the top left. The competition title 'Statoil/C-CORE Iceberg Classifier Challenge' is in bold at the top, followed by the task description 'Ship or iceberg, can you decide from space?'. The host 'Statoil' and participation stats ('3,343 teams · a month ago') are in the bottom left. The total prize money '\$50,000 Prize Money' is on the right. At the bottom right, there is additional text: 'Data and Remote Sensing expertise provided by C-CORE' with the C-CORE logo.

# ショッピングサイトの画像から多クラス分類

タスク：複数の画像が添えられた 商品ID ごとに多クラス分類

評価指標：Accuracy

The screenshot shows a product page for a Philips Senseo Original HD7818/59 - argent coffee machine. The page includes a search bar, navigation menu, and a sidebar with promotional offers. The main content features a large image of the coffee machine, its specifications, and purchase options.

**Product Information:**

- Product Name:** PHILIPS SENSEO Original HD7818/59 - argent
- Rating:** ★★★★☆ (4 stars) from 1 review
- Stock Status:** En stock !
- Delivery:** Livré dès aujourd'hui (Delivered today)
- Payment:** Réglez en 3, 5 ou 10 fois... avec la Carte Cdiscount
- Warranty:** Garantie Panne 4 ans (12,99€ soit 0,27€/mois)
- Product Details:** 1450W - Capacité: 0.7L - Pression: 1 bar - 2 tasses en même temps - Technologie SENSEO Booster d'arômes - Sélecteur d'intensité - Préparation: 30s à 1min - Câble : 0.8m - Réservoir d'eau amovible - Arrêt auto après 30min - Témoin de réservoir vide - Garantie : 2 ans

**Purchase Options:**

- Price:** 89,99 €\* **49€99** (With 0,30 € eco participation)
- Delivery Options:** Cdiscount à volonté (Delivery in express, free and unlimited for 29€ 10€ the 1<sup>st</sup> year. See CGA)
- Quantity:** 1
- Add to Cart:** Ajouter au panier
- Connect for Quick Purchase:** Connectez-vous pour activer l'achat rapide
- Store Pickup:** Retrait immédiat en magasin (55,85€)
- Other Sellers:** Autres vendeurs sur Cdiscount

**Product ID Classification:**

商品ID ごとに1～4枚の複数画像が与えらる。  
この複数画像から **5270** クラスに分類する

# データセットの基本情報をおさえる

クラス数は 5270 クラス（3階層のカテゴリが定義されている）

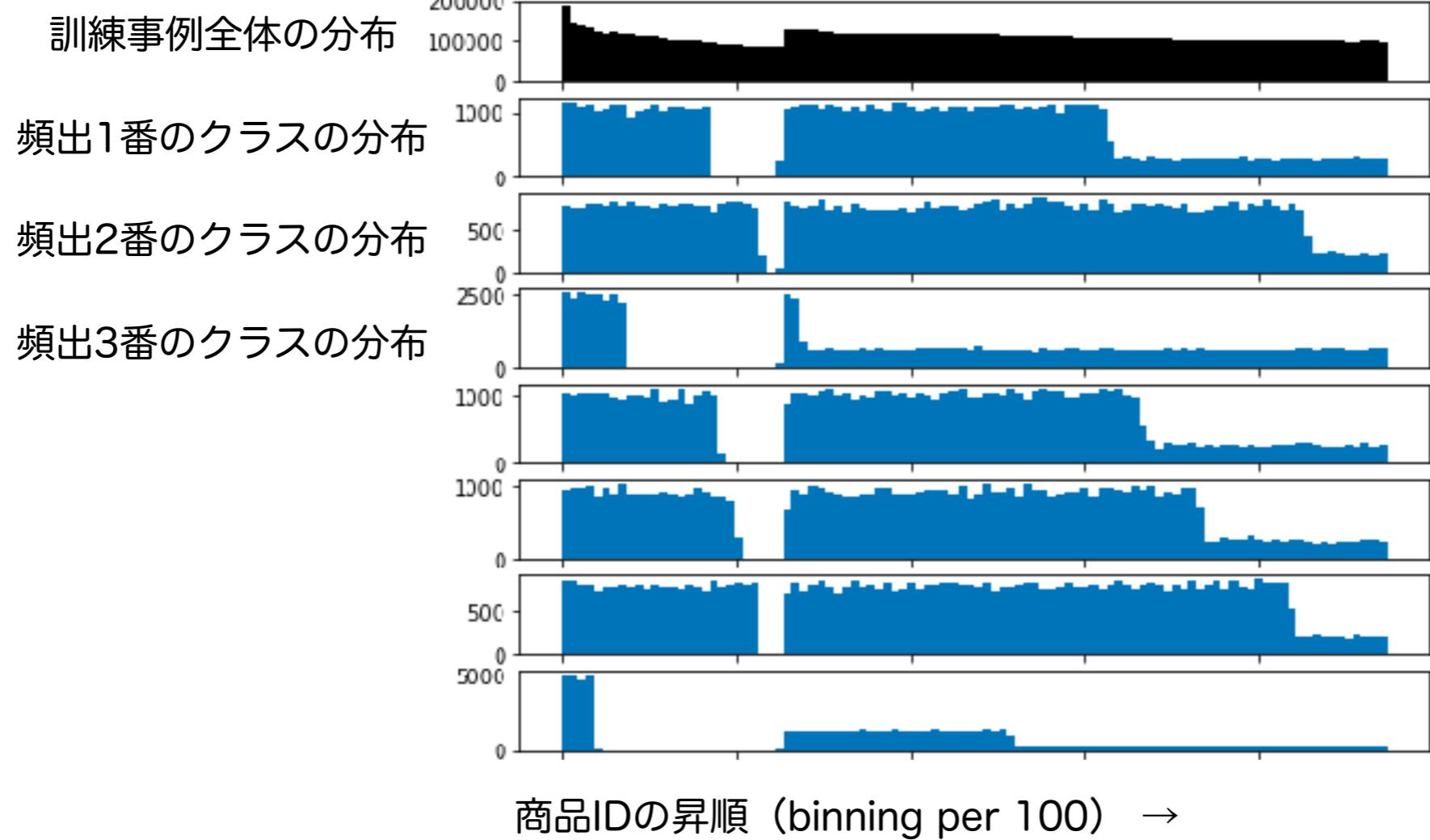
訓練事例	58.2 GB	7,069,896 商品 ( $7.0 * 1e+6$ )
テスト事例	14.5 GB	1,768,182 商品 ( $1.8 * 1e+6$ )

- 非常に skew なクラス分布である
- 仮説：商品ID は利用できないか？
- 仮説：訓練事例とテスト事例で同じ画像が存在しないか？

基本方針として CNN モデルを画像単位で入力して学習し、商品ごとに結果を平均で集約して予測する。

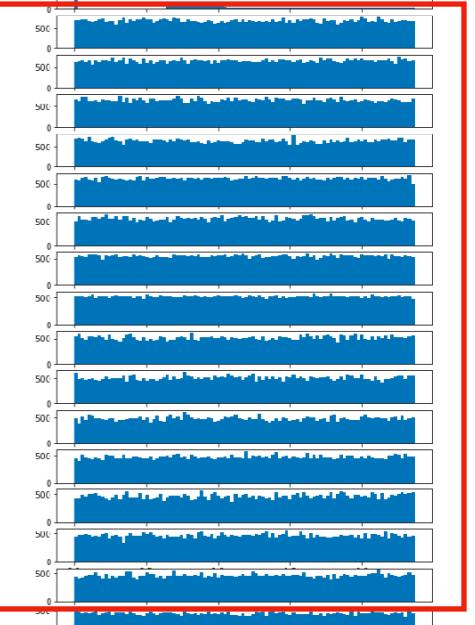
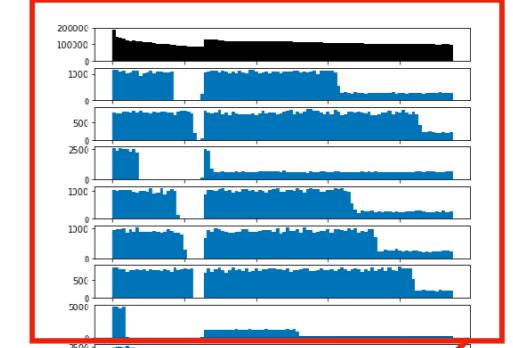


# 仮説：商品ID は利用できないか？



結論：特定のクラスだけ 商品ID に規則性がある。

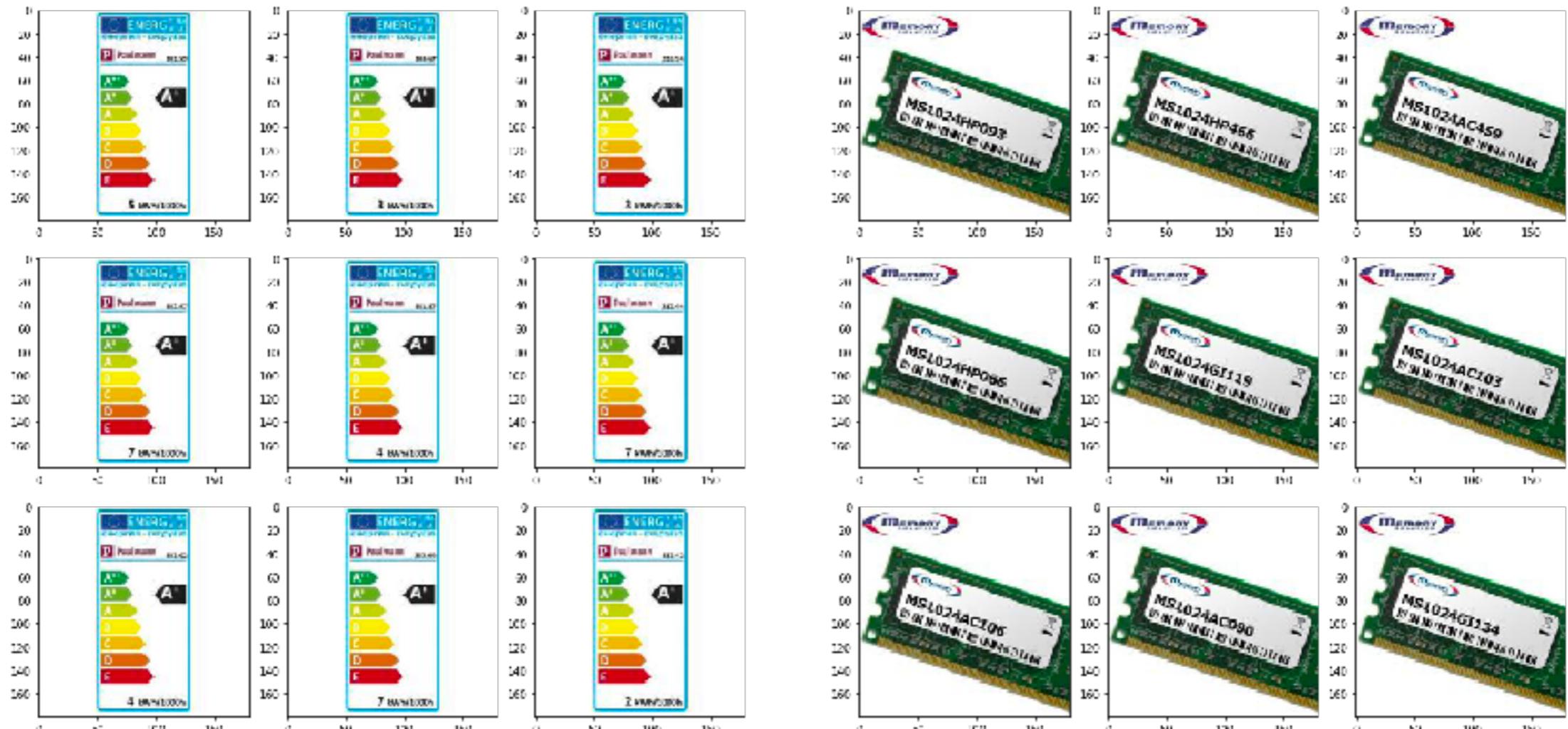
ラベルの現れない商品ID値域をブラックリストとして、  
別のラベルを答えるヒューリスティクスを使う。



# 仮説：訓練/テスト事例で同じ画像が存在

MD5をすべての画像で計算した結果、商品に対応するすべての画像が完全に一致する場合と、一部の画像だけ完全に一致する場合がある。

Hashing による類似画像も同様の結果。



同一画像

類似画像

# CNN モデル (1/3)

GPU 使用率を意識しないと、いつまで経っても訓練が終わらない。

**CPU ボトルネック:** GPU 枚数を増やすと一度に処理できる事例数が増える。すると今度は Data augmentation などの画像処理の計算コストが増える。マルチスレッドでデータを処理するなどの工夫が必要となる

**IO ボトルネック:** 大容量 SSD を購入するか、RAM (tmpfs) に BSON ファイルをデプロイするなどして解消する必要がある

BSON (binary format; MongoDB) で格納されているデータを効率的に VRAM に絶え間なく転送して CUDA に計算させつづける必要がある。

```
Every 5.0s: nvidia-smi | head -n20
```

```
Wed Feb 28 02:41:09 2018
```

NVIDIA-SMI 390.12							Driver Version: 390.12	
GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Incorr.	ECC	
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.		
0	Tesla V100-SXM2...	Off	00000000:00:17.0	Off			0	
N/A	64C	P0	250W / 300W	935MiB / 16160MiB	100%	Default		
1	Tesla V100-SXM2...	Off	00000000:00:18.0	Off			0	
N/A	59C	P0	240W / 300W	935MiB / 16160MiB	100%	Default		
2	Tesla V100-SXM2...	Off	00000000:00:19.0	Off			0	
N/A	62C	P0	253W / 300W	935MiB / 16160MiB	99%	Default		

watch -n5 nvidia-smi するなどして目視で確認。

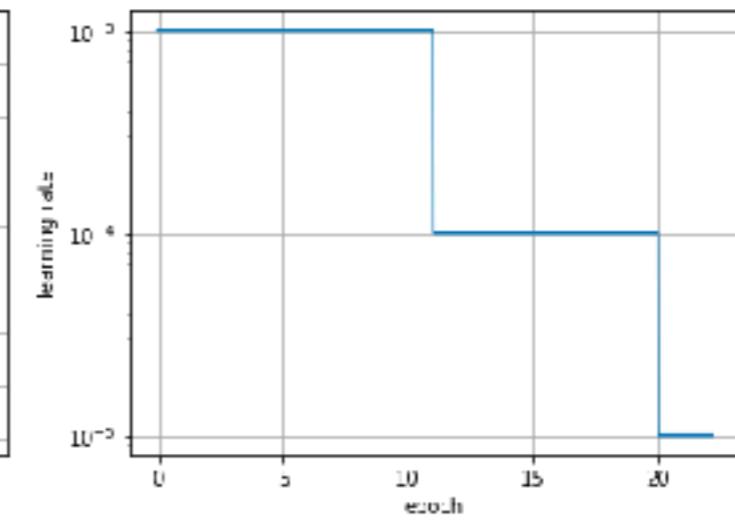
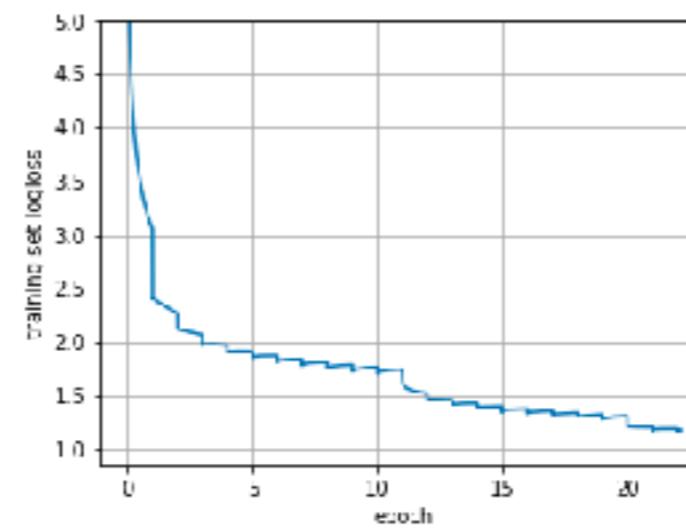
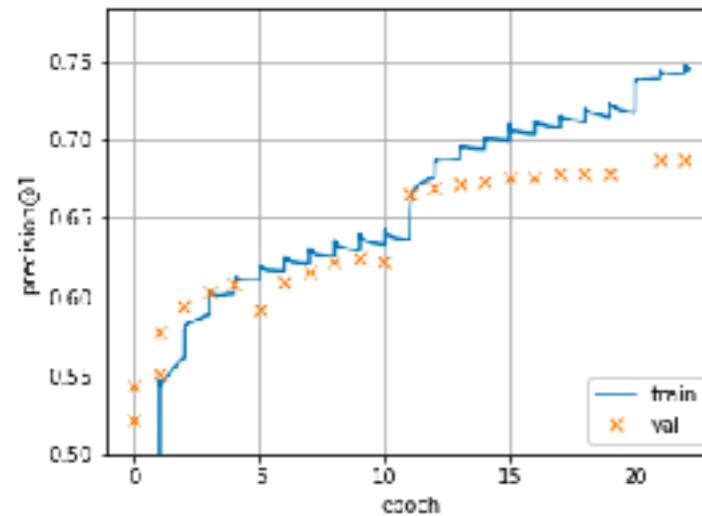
計測や確認をしないと、なかなか気がつくことが出来ない。

プログラムで IO, CPU, GPU それぞれにかかる実行時間を計測すると更に良い。

# CNN モデル (2/3)

学習について。画像枚数が非常に多いので、一回の試行に一週間待たされるこ  
ともあり、非常に難儀したところ。

- 訓練事例が ImageNet より多いので、pre-trained model の学習した特徴  
マップを壊すつもりで大きな learning rate = 1e-2 から SGD で学習
- 十分に validation set での logloss がサチったと判断できるまで learning  
rate を下げない。learning rate をすぐに下げるは後半の改善が小さい
- 実験条件が同じであれば、時間節約のため 1, 2 epoch で実験を打ち切る



# CNN モデル (3/3)

## ■ 12チャンネルCNN

1~4枚と画像の枚数が限られていたため、3 channels \* 4 の 12 channels を入力とする CNN モデルを作成した。

結果はあまり良くなかったが、アンサンブルでの diversity を上げることには成功してスコアを少しだけ押し上げた。

## ■ 専門家CNN

エラー分析したところ、いくつかの大力学カテゴリで精度が悪いため、これらに特化したモデル（専門家モデル）を用意。一部の予測結果を置き換えた。

# Single models

Best single model は Public LB で 22位 (627チーム中)。

ResNet101-12ch



Public LB: 0.72991 (52 位相当)

ResNet101



Public LB: 0.74392 (37 位相当)

DPN-92



Public LB: 0.75254 (26 位相当)

Inception V3



Public LB: 0.75226 (26 位相当)

InceptionRes V2

Expert



Public LB: 0.76047 (22 位相当)

# アンサンブル (重み付き平均)

テスト事例数が十分に大きなので、Public LB を確認しつつ調整。

ResNet101-12ch		Public LB: 0.72991 (52 位相当)	* 0.15	
ResNet101		Public LB: 0.74392 (37 位相当)	* 0.15	
DPN-92		Public LB: 0.75254 (26 位相当)	* 0.25	
Inception V3		Public LB: 0.75226 (26 位相当)	* 0.15	
InceptionRes V2	Expert		Public LB: 0.76047 (22 位相当)	* 0.3

アンサンブル = Public LB: 0.77950 (8 位相当)

# 6th: 5モデルのアンサンブル+ $\alpha$



同一 MD5 の訓練事例を使い、CNN モデルの softmax を補正する：

$$P(y = l|x) = \frac{\frac{m(l,x)}{n(x)}n(x) + s(l,x)\gamma}{n(x) + \gamma}$$

$n(x)$  … 訓練集合において、画像  $x$  の MD5 と同一の MD5 を持つ画像の数

$m(l, x)$  … 訓練集合において、画像  $x$  の MD5 と同一の MD5 を持ち、かつ  $y=l$  である画像の数

$s(l, x)$  … 予測モデルのカテゴリ  $l$  に対応する softmax output

$\gamma$  … ハイパーパラメーター。validation では  $\gamma=1$  から  $0.5$  あたりが最適

# 6th: 5モデルのアンサンブル+ $\alpha$



同一 MD5 の訓練事例を使い、CNN モデルの softmax を補正する：

$$P(y = l|x) = \frac{\frac{m(l,x)}{n(x)}n(x) + s(l,x)\gamma}{n(x) + \gamma}$$

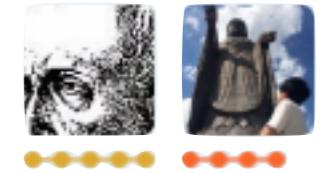
(1) 訓練事例の正例の割合 (2) Softmax  
(1), (2) の重み付き平均とみなす

$n(x)$  … 訓練集合において、画像  $x$  の MD5 と同一の MD5 を持つ画像の数

$m(l, x)$  … 訓練集合において、画像  $x$  の MD5 と同一の MD5 を持ち、かつ  $y=l$  である画像の数

$s(l, x)$  … 予測モデルのカテゴリ  $l$  に対応する softmax output

$\gamma$  … ハイパーパラメーター。validation では  $\gamma=1$  から 0.5 あたりが最適



# 6th: 5モデルのアンサンブル+ $\alpha$

同一 MD5 の訓練事例を使い、CNN モデルの softmax を補正する：

$$P(y = l|x) = \frac{\frac{m(l,x)}{n(x)}n(x) + s(l,x)\gamma}{n(x) + \gamma}$$

(1) 訓練事例の正例の割合 (2) Softmax  
(1), (2) の重み付き平均とみなす

$n(x)$  … 訓練集合において、画像  $x$  の MD5 と同一の MD5 を持つ画像の数

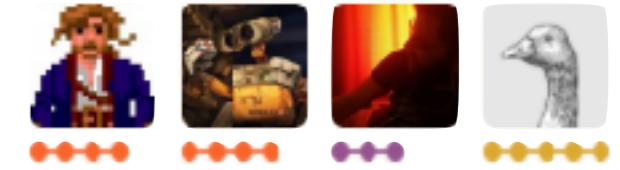
$m(l, x)$  … 訓練集合において、画像  $x$  の MD5 と同一の MD5 を持ち、かつ  $y=l$  である画像の数

$s(l, x)$  … 予測モデルのカテゴリ  $l$  に対応する softmax output

$\gamma$  … ハイパーパラメーター。validation では  $\gamma=1$  から 0.5 あたりが最適

+ クラス分布から計算したブラックリストのヒューリスティクスを適用

0.77950 (8位相当) → 0.78315 (6位相当)



## 7th: 特徴抽出して集約&更に学習

ベースモデルの bottleneck block の出力を使って商品単位に集約&学習。OCR (EAST [14] + CRNN [12] ) してテキスト情報を抽出。

これらを入力として MLP, RNN, NetVlad [13] などのトップレベルモデルを学習。最後にベースモデルとトップレベルモデルでアンサンブル。

- ベースモデルは InceptionResnetv2, Resnet101, SE-InceptionV3, Xception
- 4枚の画像を入力して得た Bottleneck features を商品ごとにグループして トップレベルモデルの入力とする
- NetVlad の Pooling 学習と Gating は、過去コンペ “Youtube 8M Kaggle Large-Scale Video understanding challenge” の解法でも使われ、LOUPE としてツールボックスが公開されている



## 5th: ニ值分類問題に変換

ありえそうな予測クラスの候補 “possible\_class\_id” という特徴量を作り、多クラス分類問題を **ニ值分類問題に変換**。そして XGBoost などで学習。

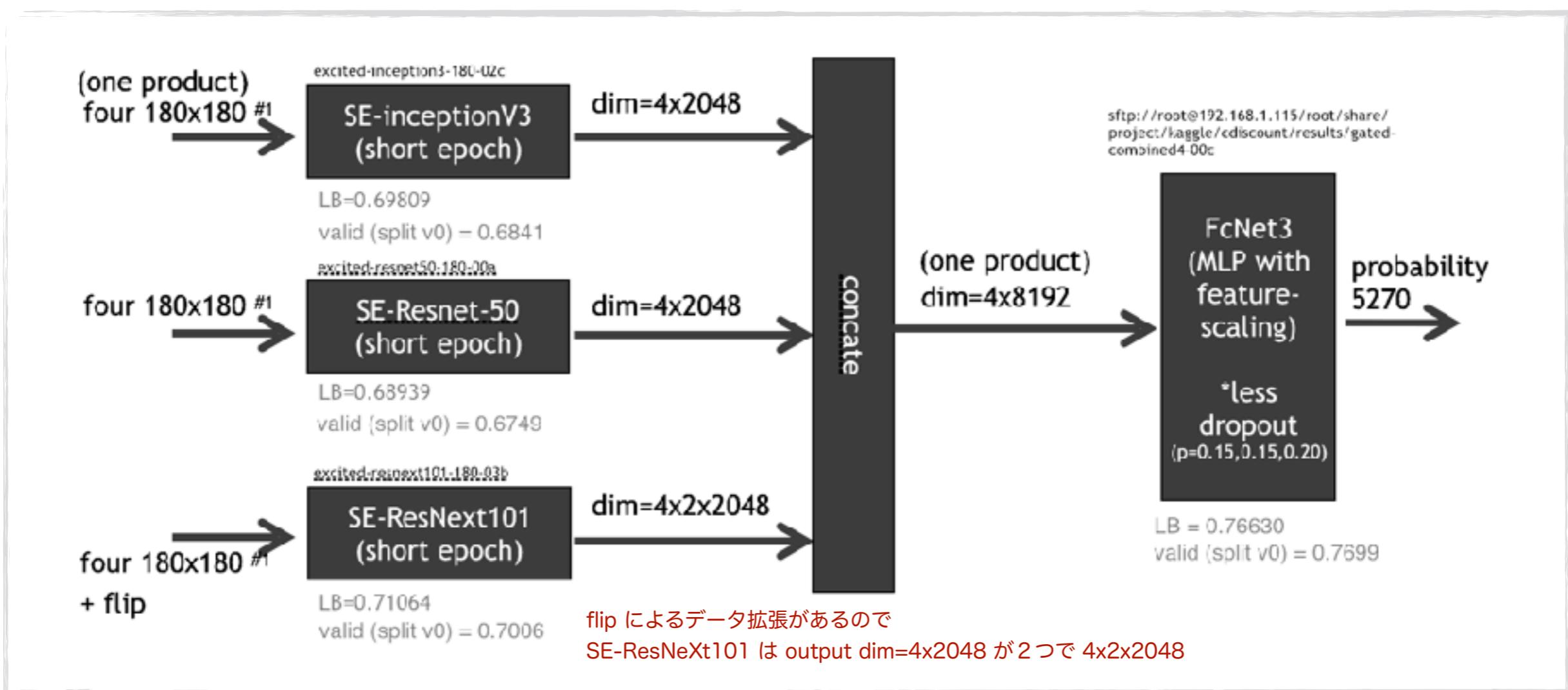
- MD5 レベルで一致する画像の最頻出クラスIDをカテゴリカル特徴量とした
- MD5 レベルで一致する画像の重複数を量的特徴量とした

複数枚の画像を商品ごとにまとめた集約のハンドリングと、MD5 レベルの一致の情報をモデルに組み込んでいる。



# 2nd: MLP \w gated scaling

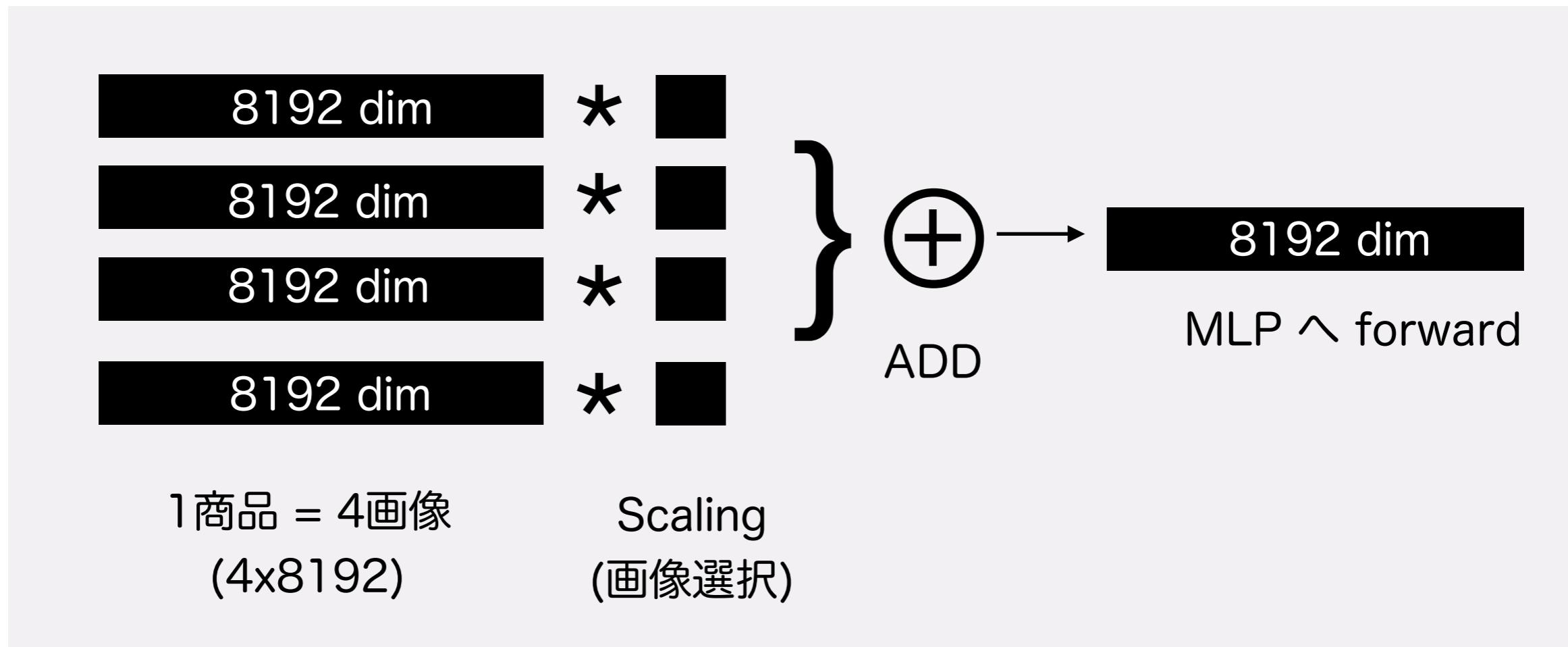
- bottleneck block の conv  $1 \times 1 \times 2048$  から 4 画像分  $4 \times 2048$  を抽出
- 各 streams から特徴を concat した  $4 \times 8192$  を入力として MLP で学習
- Multiview pooling [11] に近いが、データが大きいので学習は 2stage に分割





## 2nd: MLP \w gated scaling

- 中間成果物：画像枚数×特徴数 = 4x8192。これに gated scaling + MLP 適用
- 1画像につき 8192 の dim から SEScale module で画像を選択して加算する
- 4画像のうち1枚だけが重要という事例があるので有効そう (ref: PointNet [10])





# 2nd: MLP \w gated scaling

- 中間成果物：画像枚数×特徴数 = 4x8192。これに gated scaling + MLP 適用
- 1画像につき 8192 の dim から SEScale module で画像を選択して加算する
- 4画像のうち1枚だけが重要という事例があるので有効そう (ref: PointNet [10])

```
class FcNet3(nn.Module):    in_shape = 8192
    def __init__(self, in_shape=1000, num_classes=5270):
        super(FcNet3, self).__init__()
        self.num_classes = num_classes
        in_channels = in_shape

        self.scale = SEScale(in_channels, in_channels//2)
        self.linear1 = nn.Linear(in_channels, 7168)
        self.relu1 = nn.PReLU()
        self.linear2 = nn.Linear(7168, 4096)
        self.relu2 = nn.PReLU()
        self.fc = nn.Linear(4096, num_classes)

    def forward(self, x):
        # print('input ', x.size())
        N,V,C = x.size()
        x = F.dropout(x, p=0.50, training=self.training)
        x = self.scale(x) * x
        x = x.sum(dim=1)
        x = self.linear1(x) #; print('linear1 ',x.size())
        x = self.relu1(x)
        x = F.dropout(x, p=0.30, training=self.training)
        x = self.linear2(x) #; print('linear2 ',x.size())
        x = self.relu2(x)
        x = F.dropout(x, p=0.30, training=self.training)

        x = self.fc(x)
        return x #logits
```

```
##-----
class SEScale(nn.Module):
    def __init__(self, channel, reduction=16):
        super(SEScale, self).__init__()
        self.fc1 = nn.Linear(channel, reduction)
        self.fc2 = nn.Linear(reduction, channel)

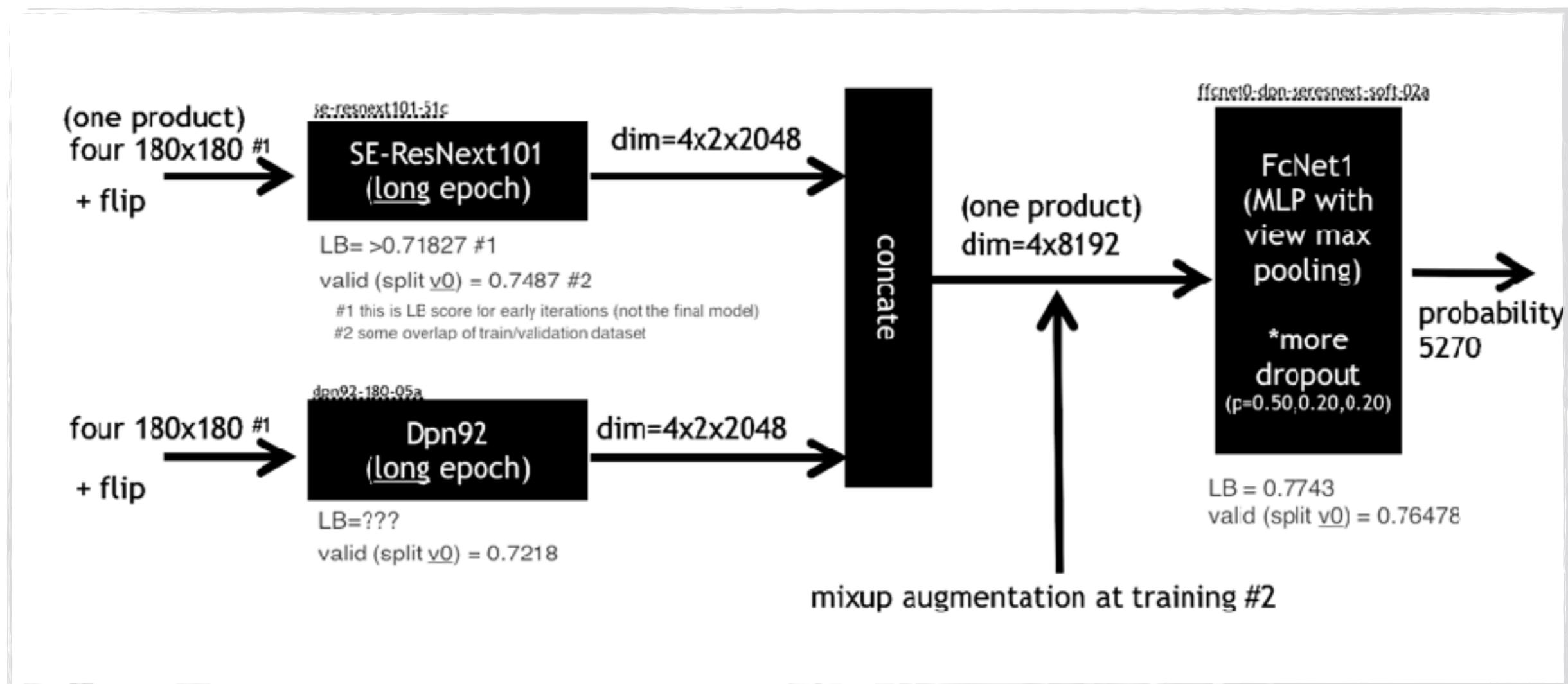
    def forward(self, x):
        x = self.fc1(x)
        x = F.relu(x, inplace=True)
        x = self.fc2(x)
        x = F.sigmoid(x)
        return x
```

N = batch size = 4096 (very large!)  
V = num of images per product = 4  
C = concat feature dim = 8192



# 2nd: MLP \w gated scaling

- ほかにも、中間層の concat 部分で Mixup augmentation [3] するモデルなど
- それぞれ微妙に異なるトップレベルモデル計4個をアンサンブル



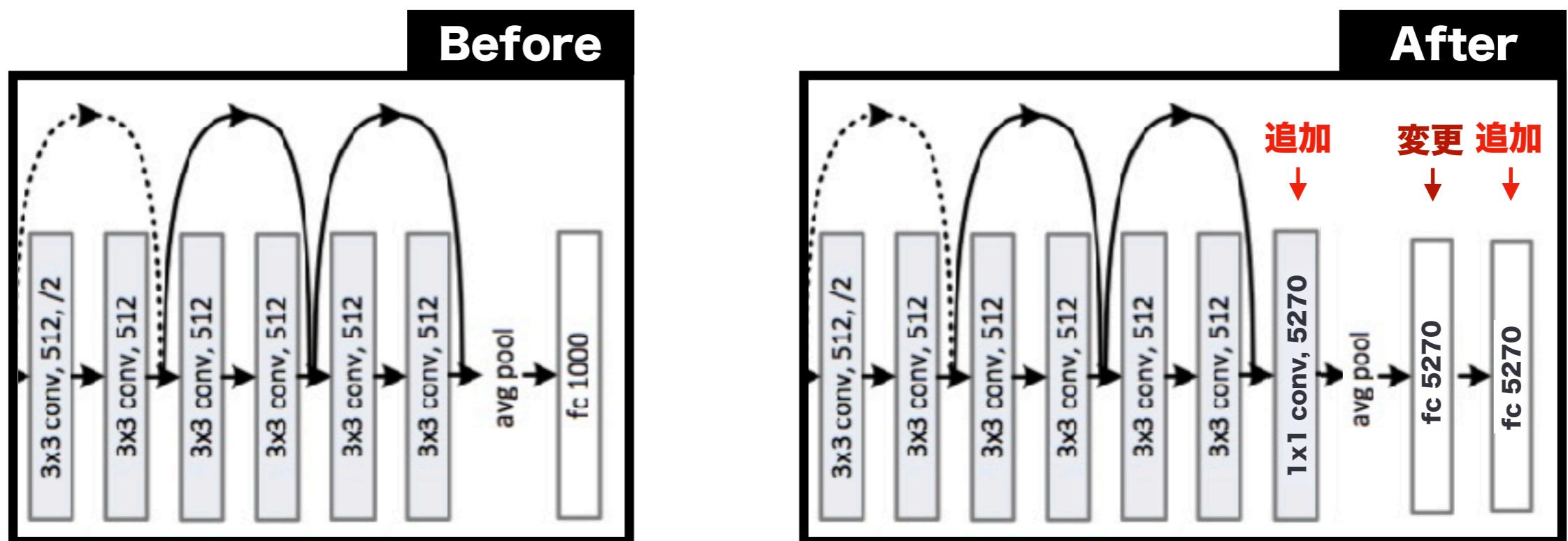
# 1st: 表現力を上げるために arch 変更



最初は小さな ResNet34 で実験をくり返してモデルの検証。

多くの ImageNet pre-trained は 1000 クラスの設計だけど今回はクラス数は 5270 クラス。アーキテクチャの表現力に限界があり、これを解消する変更を追加。

チャンネル数を増やすために  $1 \times 1$  conv layer を追加してチャンネル数を 512 から 5270 に増やす。VGG net に習い fc layer も追加する。



# 1st: 複数画像を連結して单一画像に

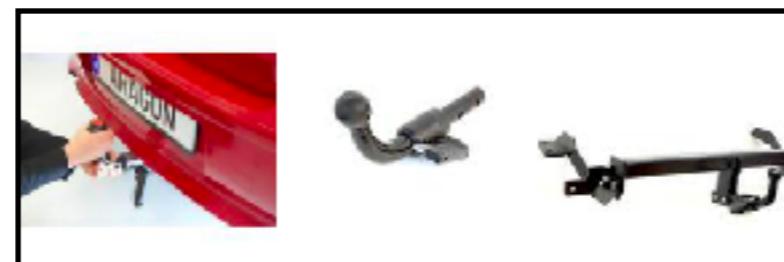


複数画像の取り扱い：連結して一つの画像にする

商品ごとの画像枚数でデータセットを分割（4通り）。画像枚数ごとのデータセットを作成して、**单一画像で学習したモデルから fine-tuning**



1商品4枚の部分セット



1商品3枚の部分セット



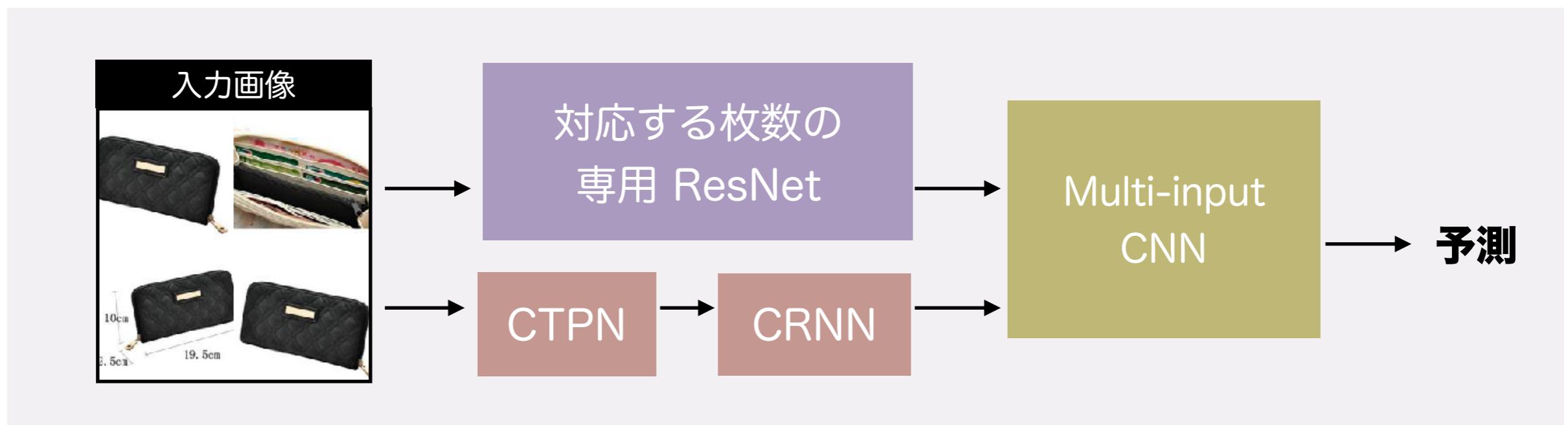
1商品2枚の部分セット

# 1st: OCR して multi-input CNN



CTPN [9] でテキスト領域の抽出、CRNN [12] でテキストの抽出を行う。Multi-input CNN で CRNN features と ResNet の FC features を連結する。

中間ファイルは uint8 に量子化 (\* 255 して cast) して sparse matrix として保存する。+400MB が 40MB 以下になる。



[9] Detecting Text in Natural Image with Connectionist Text Proposal Network

[12] An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition

# 最近のコンテストにおける解法の解説

2つのコンテストにおける解法を紹介する。

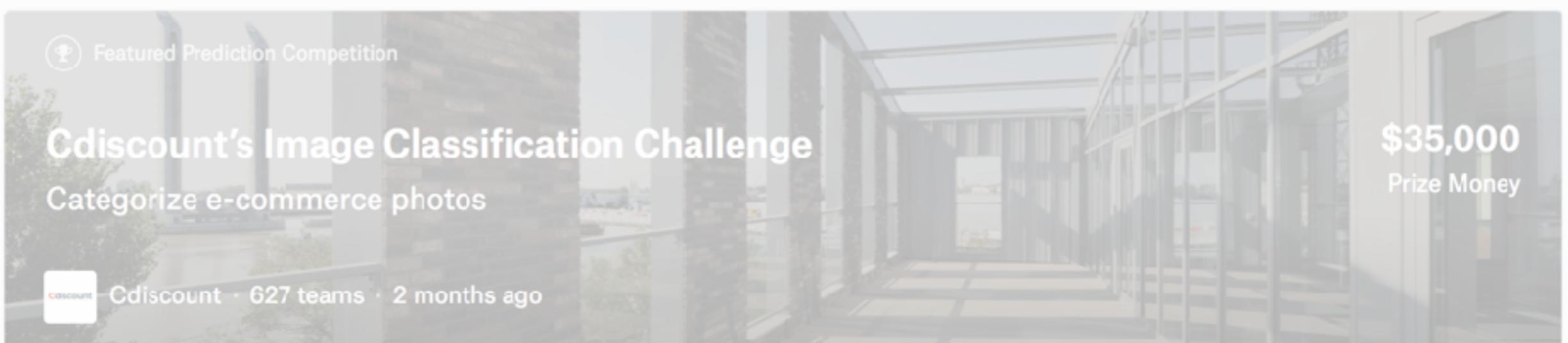
Featured Prediction Competition

## Cdiscount's Image Classification Challenge

Categorize e-commerce photos

 Cdiscount · 627 teams · 2 months ago

\$35,000 Prize Money



Featured Prediction Competition

## Statoil/C-CORE Iceberg Classifier Challenge

Ship or iceberg, can you decide from space?

 Statoil · 3,343 teams · a month ago

~ 2018/01/23

\$50,000 Prize Money

Data and Remote Sensing expertise provided by





# Statoil/C-CORE Iceberg

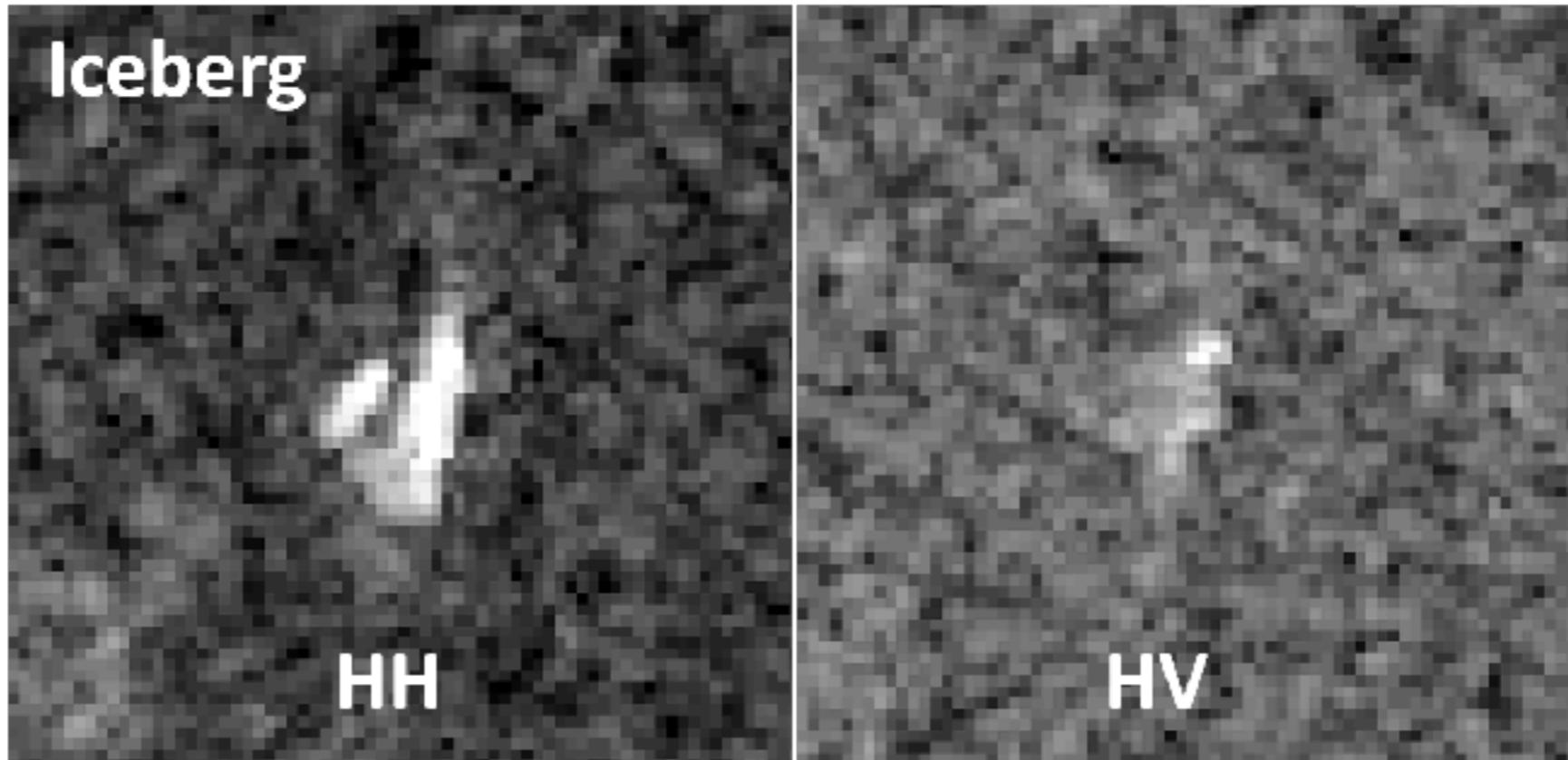
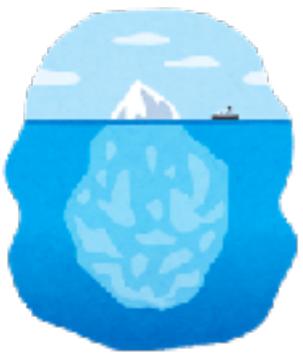
タスク：合成開口レーダー(SAR) の観測データから氷山か船舶か二値分類する

評価指標：Log loss

	偏向波の観測値 band_1	偏向波の観測値 band_2	id	inc_angle	is_iceberg
0	[-27.878360999999998, -27.15416, -28.668615, ...]	[-27.154118, -29.537888, -31.0306, -32.190483, ...]	dfd5f913	43.9239	0
1	[-12.242375, -14.92030499999999, -14.920363, ...]	[-31.506321, -27.984554, -26.645678, -23.76760...]	e25388fd	38.1562	0
2	[-24.603676, -24.603714, -24.871029, -23.15277...]	[-24.870956, -24.092632, -20.653963, -19.41104...]	58b2aaa0	45.2859	1
3	[-22.454607, -23.082819, -23.998013, -23.99805...]	[-27.889421, -27.519794, -27.165262, -29.10350...]	4cfc3a18	43.8306	0
4	[-26.006956, -23.164886, -23.164886, -26.89116...]	[-27.206915, -30.259186, -30.259186, -23.16495...]	271f93f4	35.6256	0

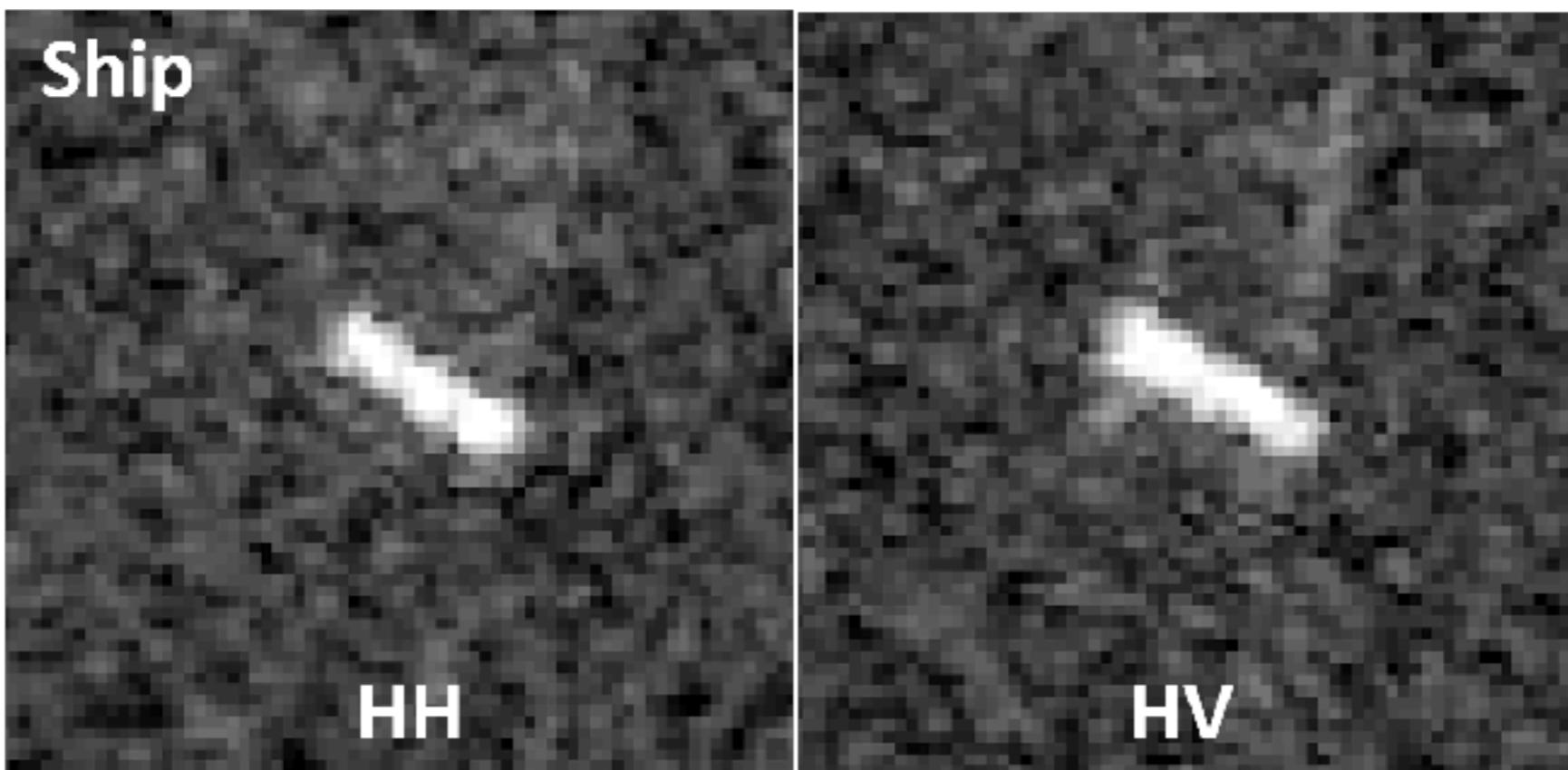
**Iceberg**

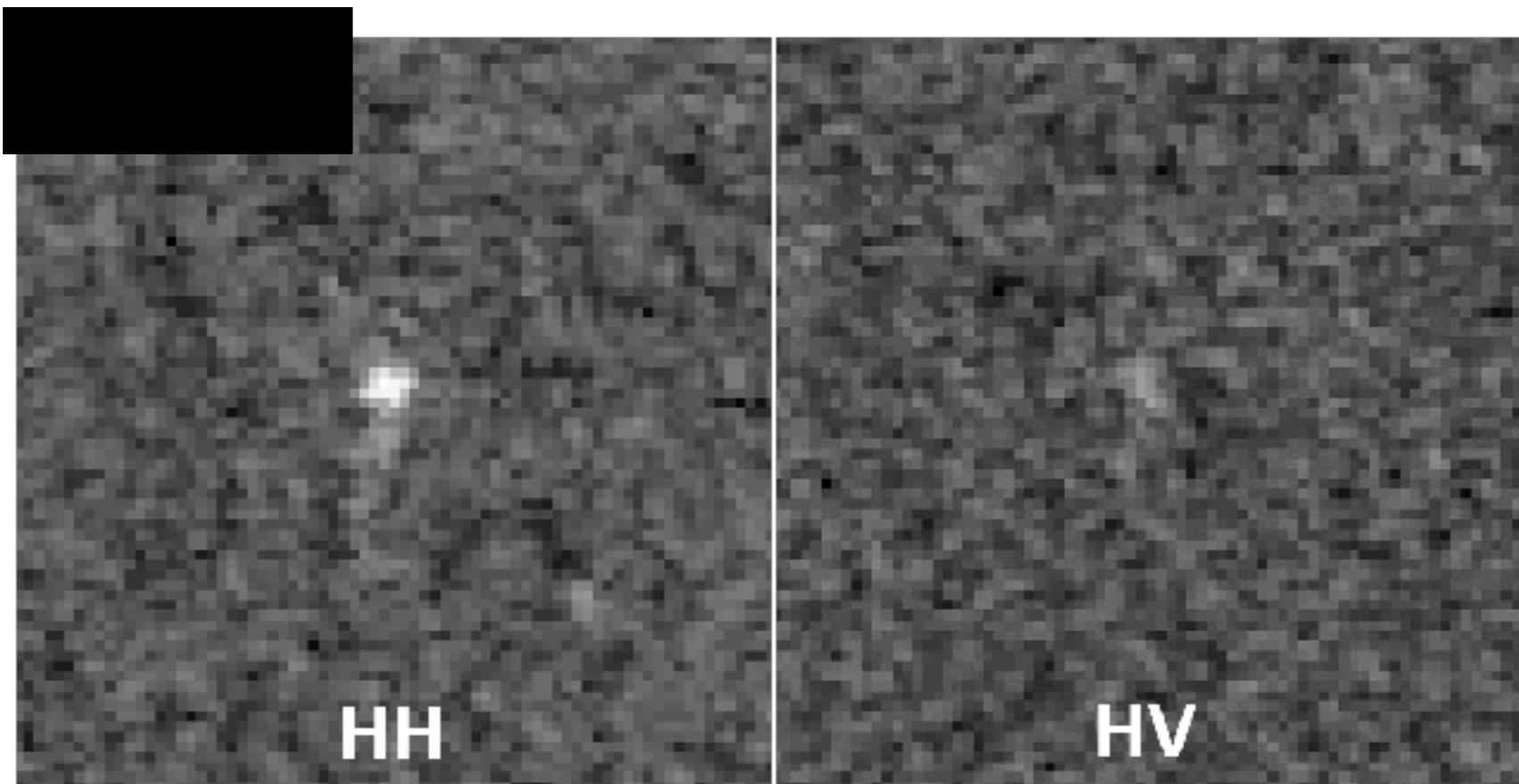
冰山



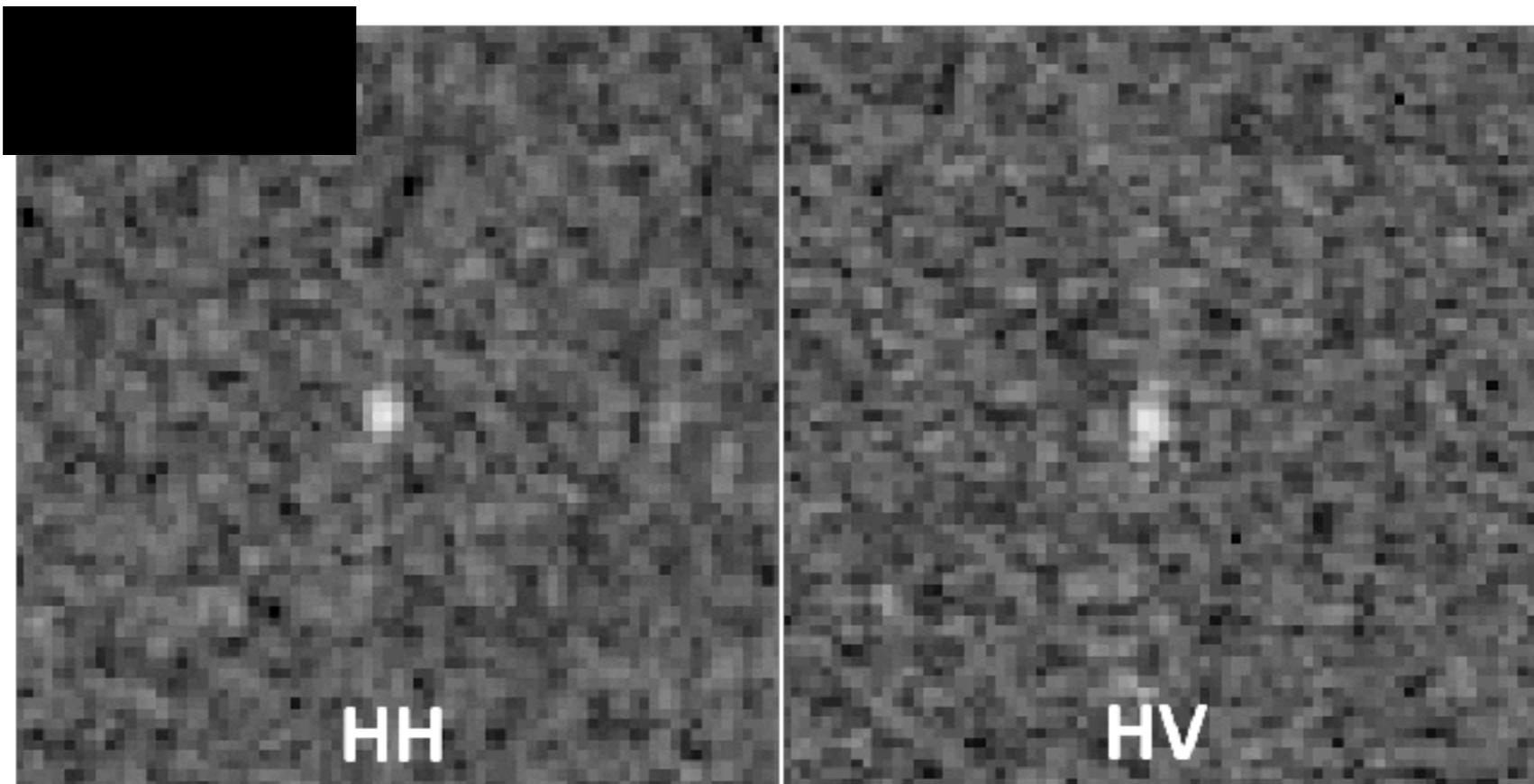
**Ship**

船舶



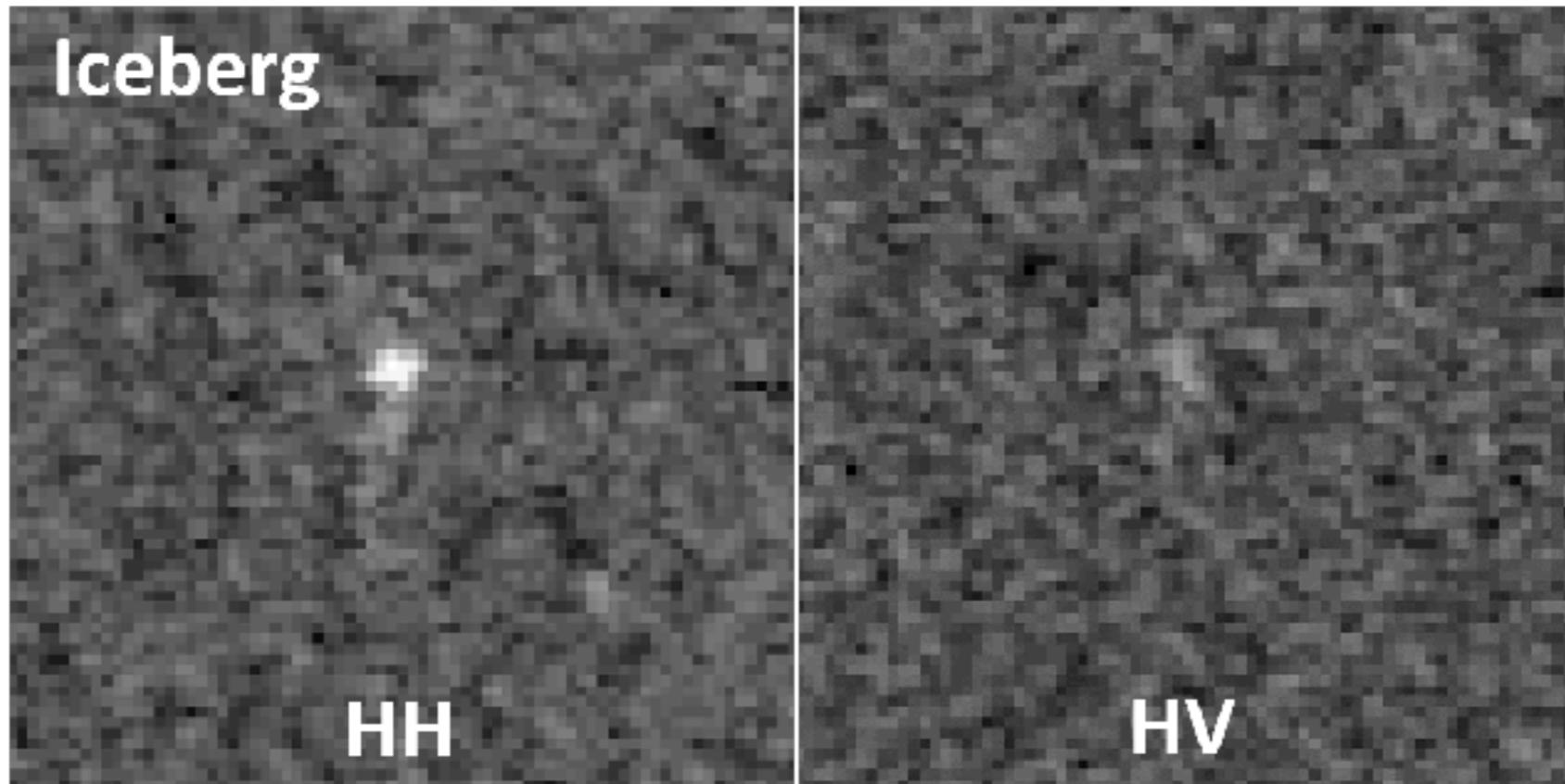
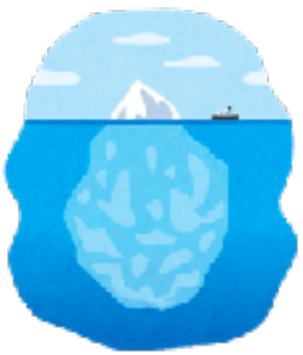


どちらが流水？



**Iceberg**

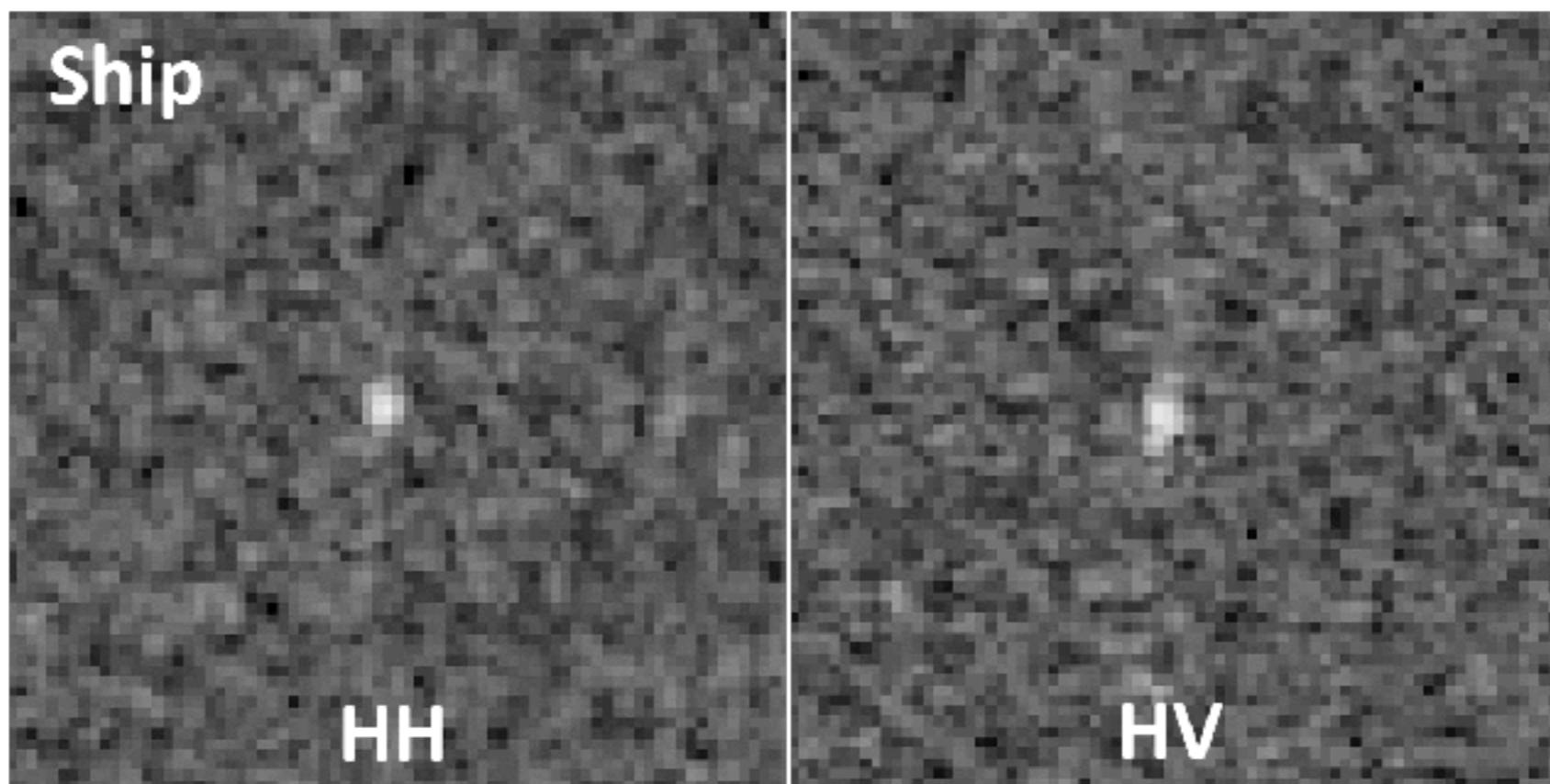
氷山



困難な事例がある。  
過学習に注意が必要。

**Ship**

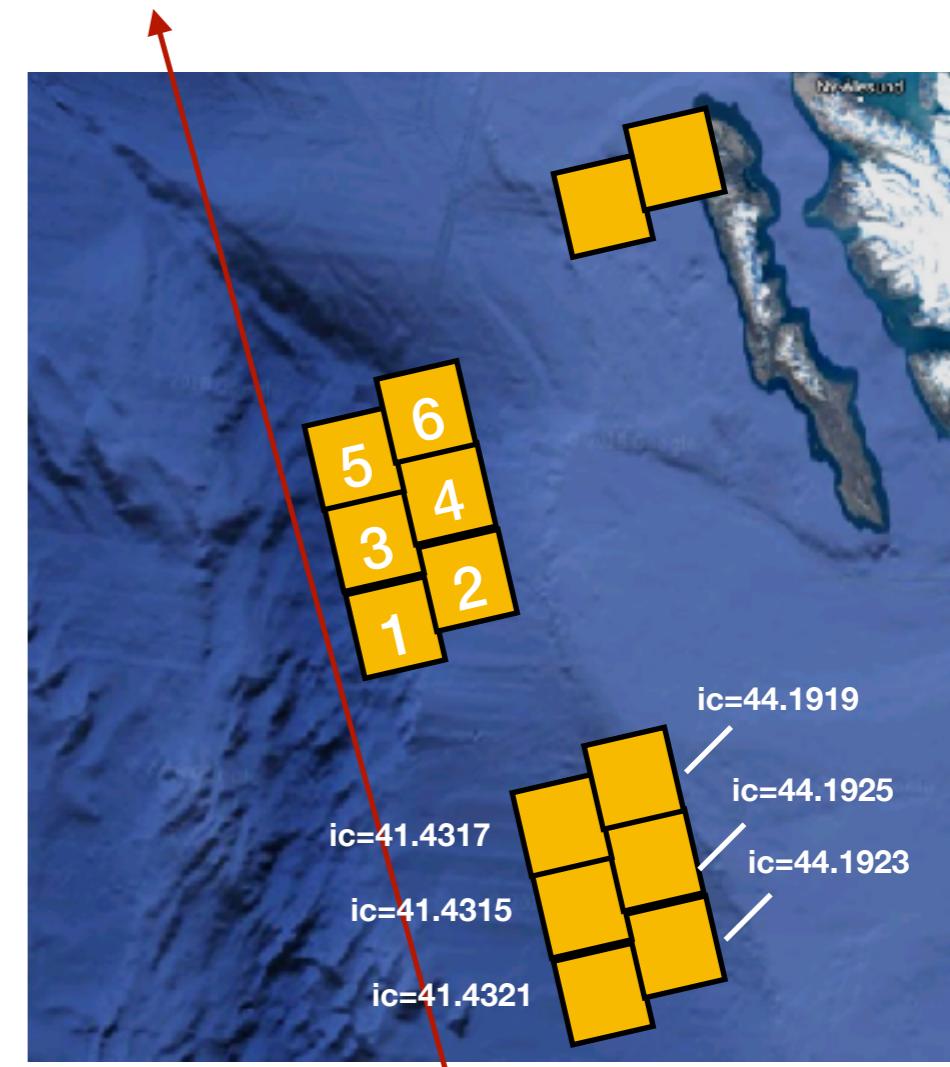
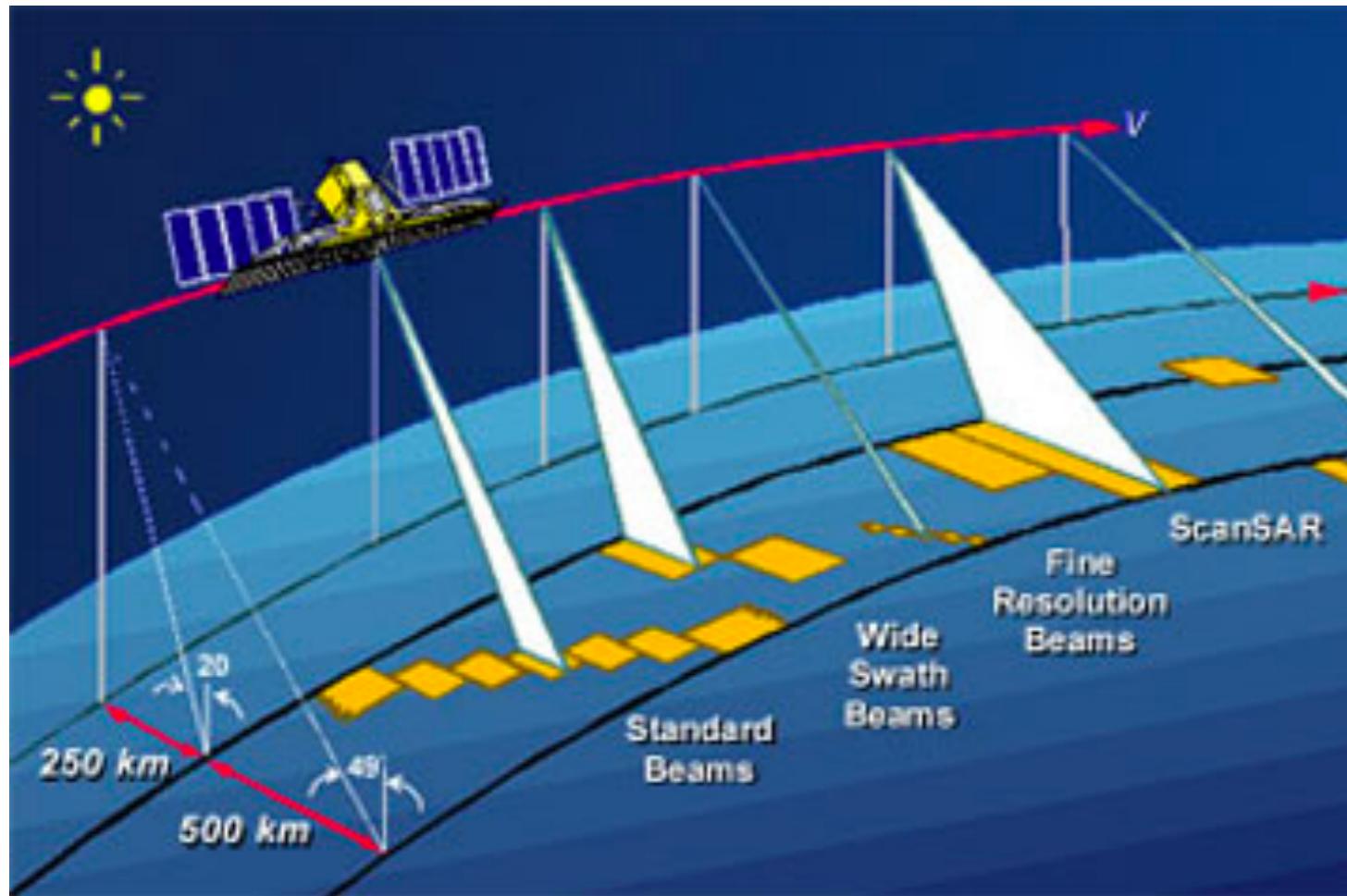
船舶



# データに対する理解を深める

合成開口レーダーは入射角（Incidence angle）を調整して観測するので、ほぼ同値のグループができると考えることが出来る。

（※ 素人による考察なので、間違いがあるかもしれません）

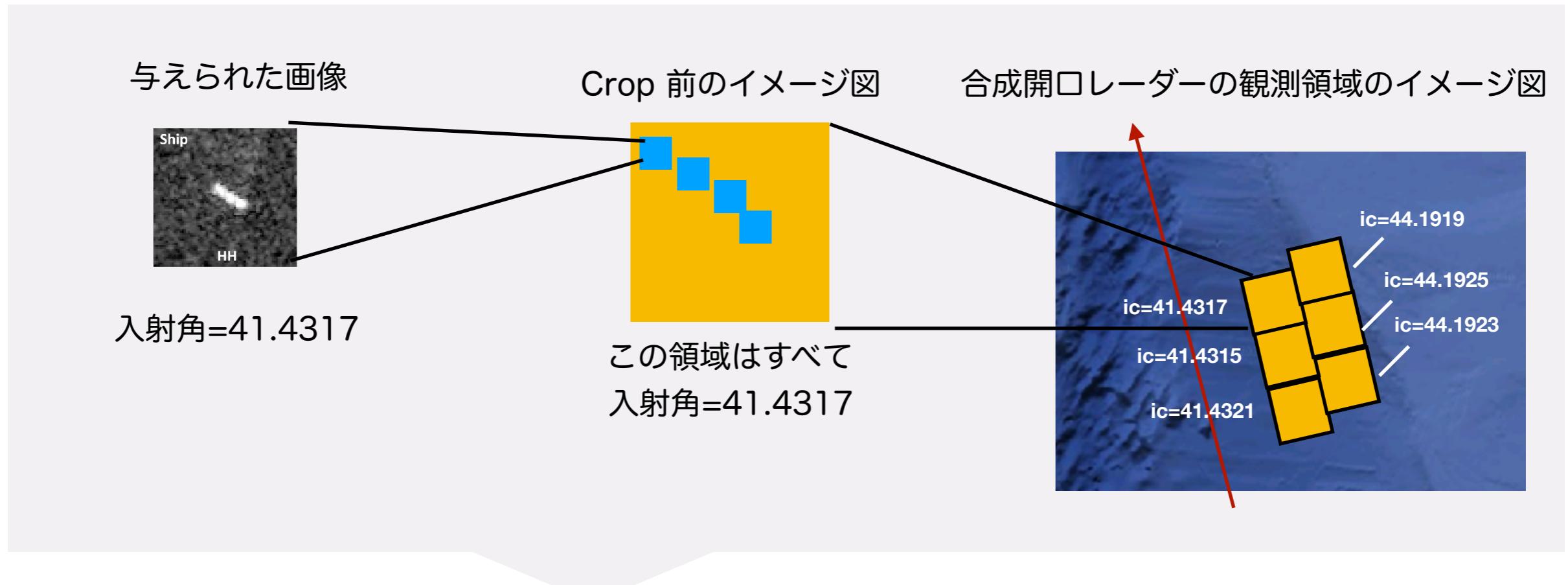


出典（左）：<http://www.asc-csa.gc.ca/eng/satellites/radarsat1/components.asp>

（右）Imagery © 2018 IBCAO, Landsat / Copernicus, Map data © 2018 Google

# データセット画像はあらかじめ crop したもの

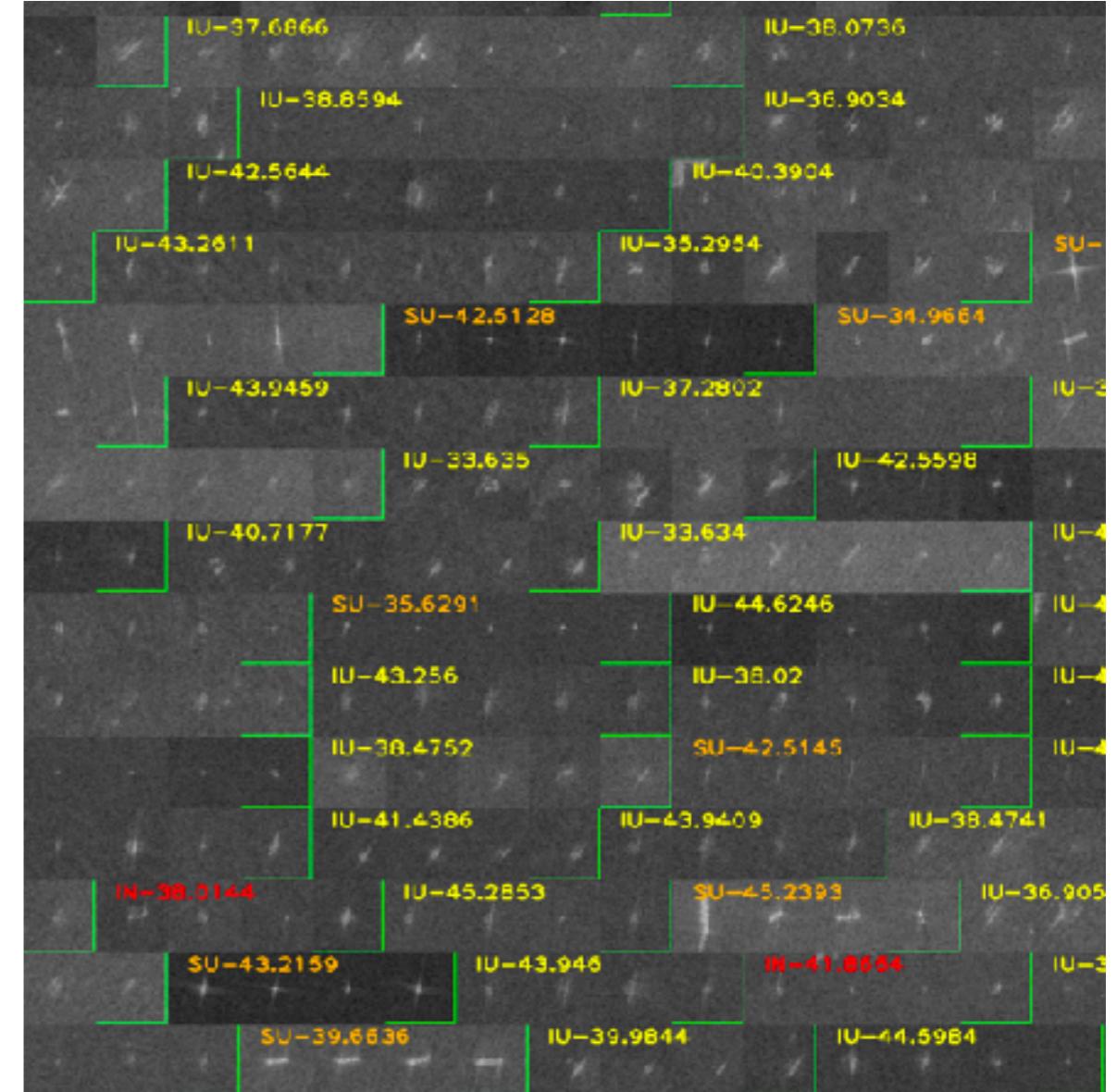
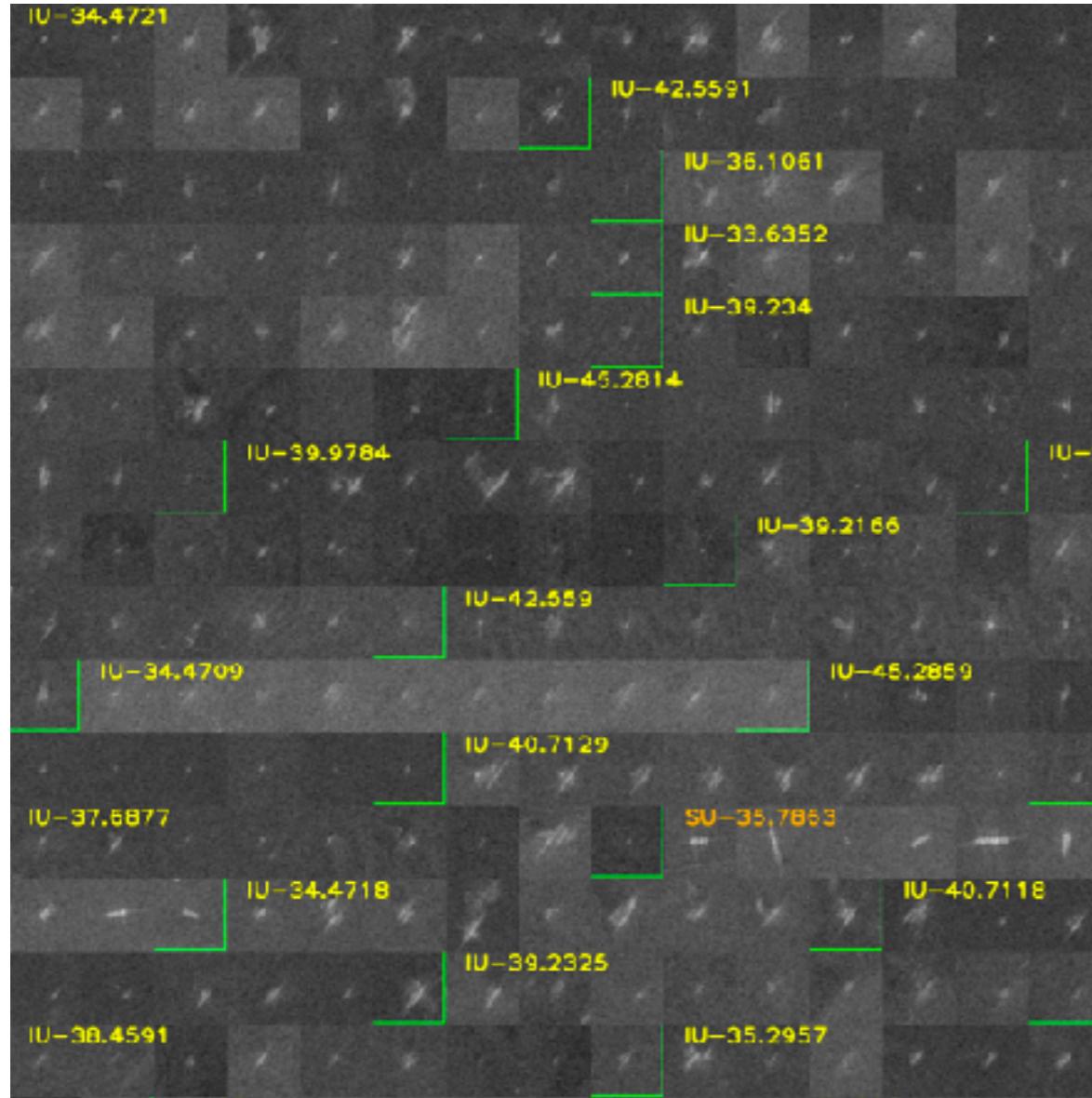
与えられた観測データは、船舶あるいは氷山が中心らしきものが写っている。あらかじめ crop された画像。



仮説：入射角の少数表示桁すべてが完全一致する場合、  
元々が同じ観測領域 ■ から切り取った crop された画像 ■ で与えられる

# 仮説: 入射角が一致するグループは同じラベル

黄色=すべて氷山 オレンジ=すべて船舶 赤=氷山と船舶が混ざっている



入射角が完全一致する事例グループは、ほぼ“おなじラベル”になる。  
背景を考えると妥当な結果。

# ヒューリスティクスで訓練事例を追加

「学習した CNN モデル」の Accuracy よりも  
「訓練事例にある、完全に同一の入射角の観測事例」  
を探して最頻出ラベルを答えたほうが Accuracy 良い (+97%)。

訓練事例が少ないので、上の最頻出ラベルをすべて訓練事例に追加した。

訓練事例数 : 1604 → 3546

テスト事例数 : 8424 → 3425

実際にはダミー画像で水増しされているので評価の対象となるテスト事例は  
とても少ない。テスト事例も少ないので、Public LB を参考にするのは危険

# CNN モデルの作成

- ・ 画像だけを使ったモデルを用意する（量的変数を同時に学習しない）
- ・ 訓練事例数が少ないので 10-fold CV でモデルを評価＆チューニング
  - ・ モデルは 4層 Convolution Layer だけで十分そう（変えても大差ない）
  - ・ pre-trained model を使うと逆に悪くなる

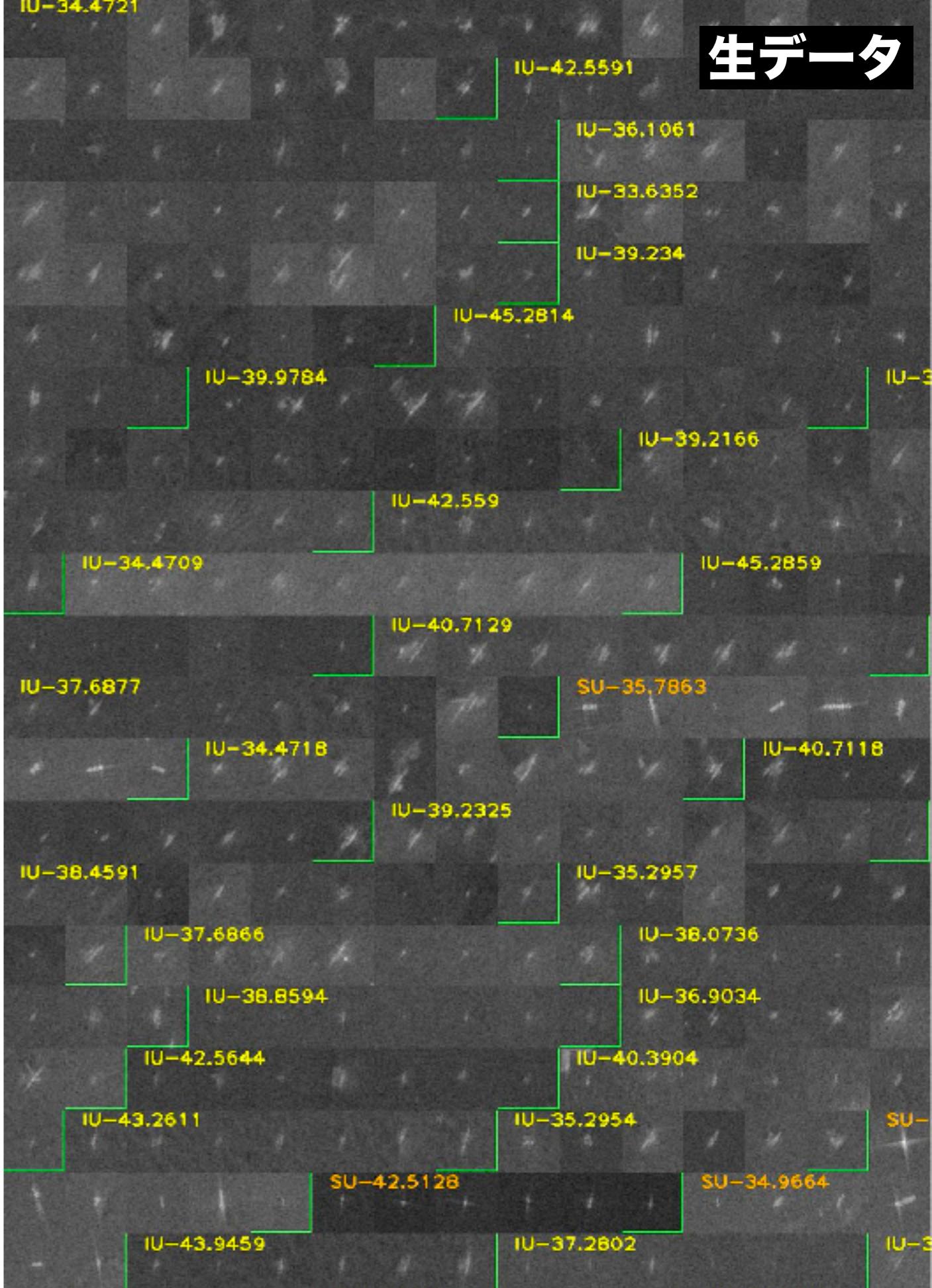
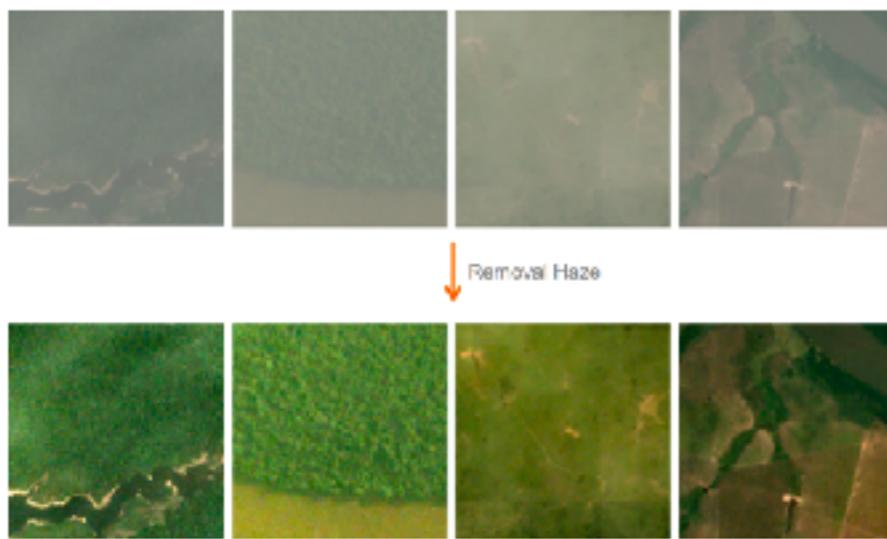
	前処理	CV	Public LB
4L CNN	Scaling	0.2212	
4L CNN	Raw	0.2191	
ResNet50	Scaling	0.2206	
VGG16	Scaling	<b>0.2107</b>	Public LB は確認していない

# 入射角ごとに正規化

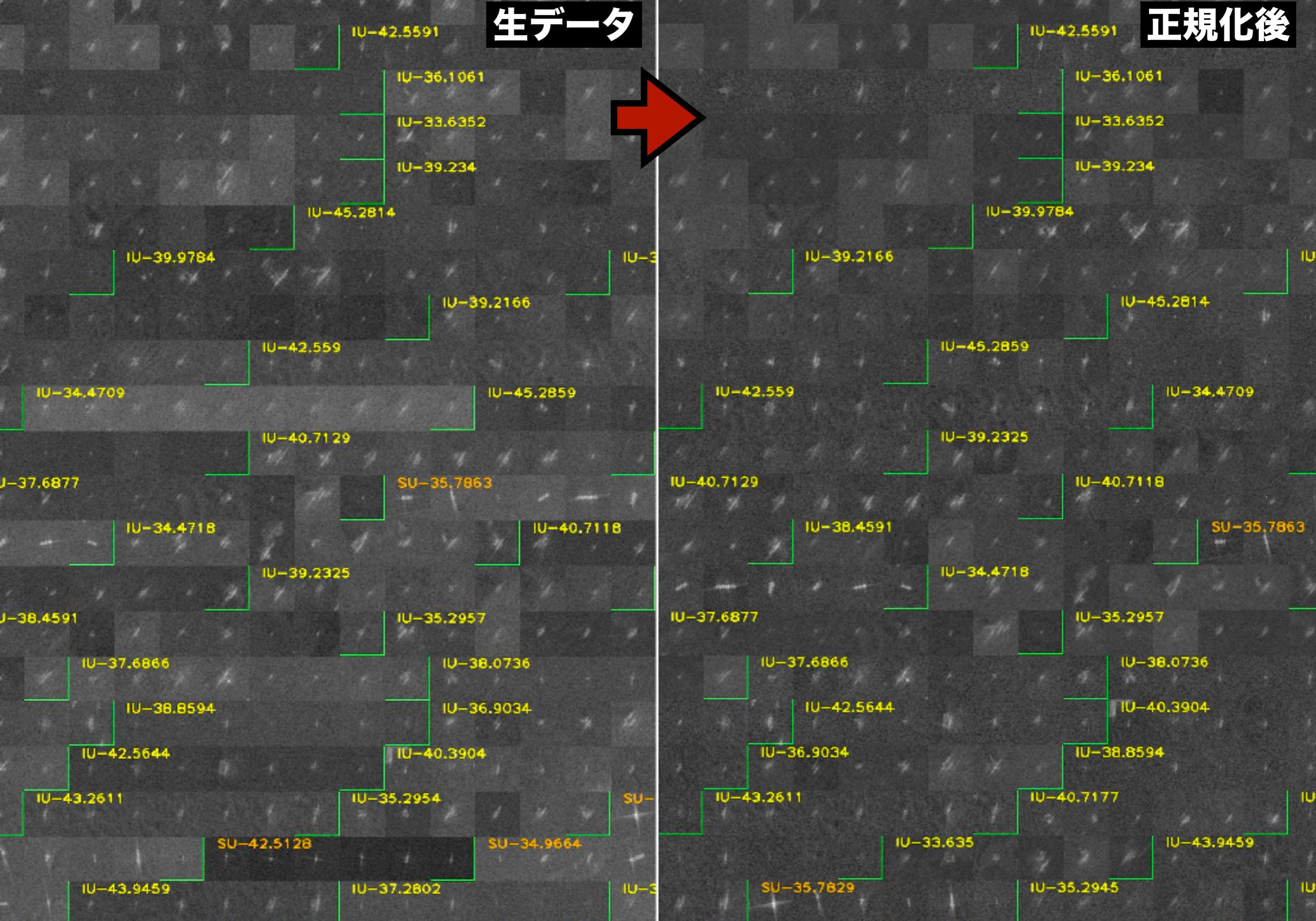
入射角ごとに傾向が別れている。

グループ化して統計量を計算して、  
グループごとに正規化する。

過去コンペの優勝者が霧のある画像を  
霧除去[6]により正規化した解法から  
着想を得た。正規化は重要。



JU-34.4721



# 入射角ごとのデータ正規化で改善

入射角を使ったデータ正規化を行と CV 改善

Data Augmentation (rot, flip, zoom) も改善に有効であった

	前処理	CV	Public LB
4L CNN	Scaling	0.2212	
4L CNN	Norm+ Scaling	0.1824	
4L CNN +DA	Norm + Scaling	<b>0.1615</b>	

Public LB は確認していない

# 5th: 入射角の情報をモデルに組み込む



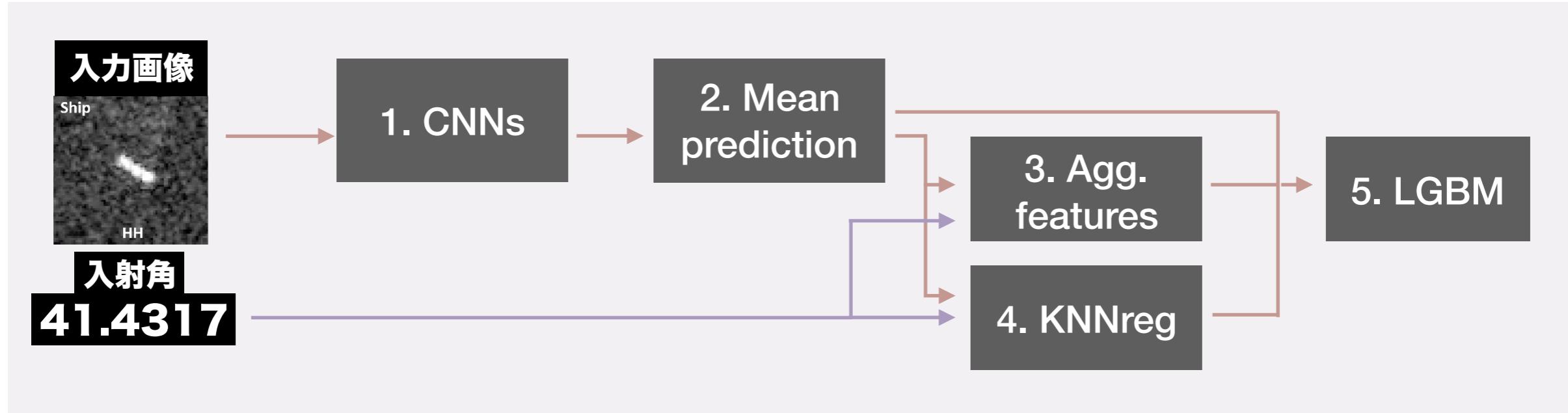
量的変数（入射角）の情報をモデルに組み込むために、CNN モデルの予測結果と量的変数を stacking した。

16 個の CNN モデル + 量的変数 x3 で LightGBM と deepbit (GBDT) で学習。最後に重み付き平均。

	前処理	CV	Public LB
4L CNN	Scaling	0.2212	
4L CNN	Norm+Scaling	0.1824	
4L CNN +DA	Norm+Scaling	0.1615	
stage2 (LGBM)	NA	0.0895	<b>0.0873</b>
stage3 (Average)	NA	—	<b>0.0852</b>

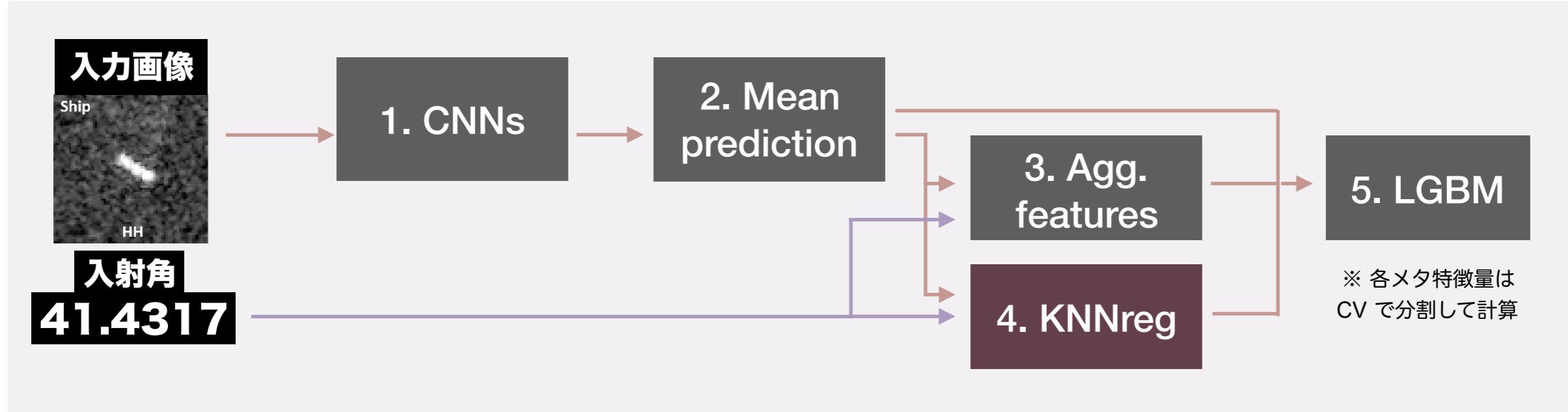
Public LB: 2nd / Private LB: 5th

# 4th: より少ないモデル数 + 入射角k近傍



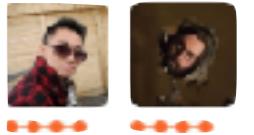
1. 5個の NN モデルを作成。5 fold CV でチューニング。
2. CV で学習したモデルの mean prediction を特徴量とする (量的変数)
3. 入射角でグループ化して集計 (mean, median, count) して特徴量作成
4. 入射角と (2) の mean prediction で KNN regressor (量的変数)
5. LGBM を (2, 3, 4) の特徴量で学習。合計で 5 つの特徴量。

# 4th: より少ないモデル数 + 入射角k近傍

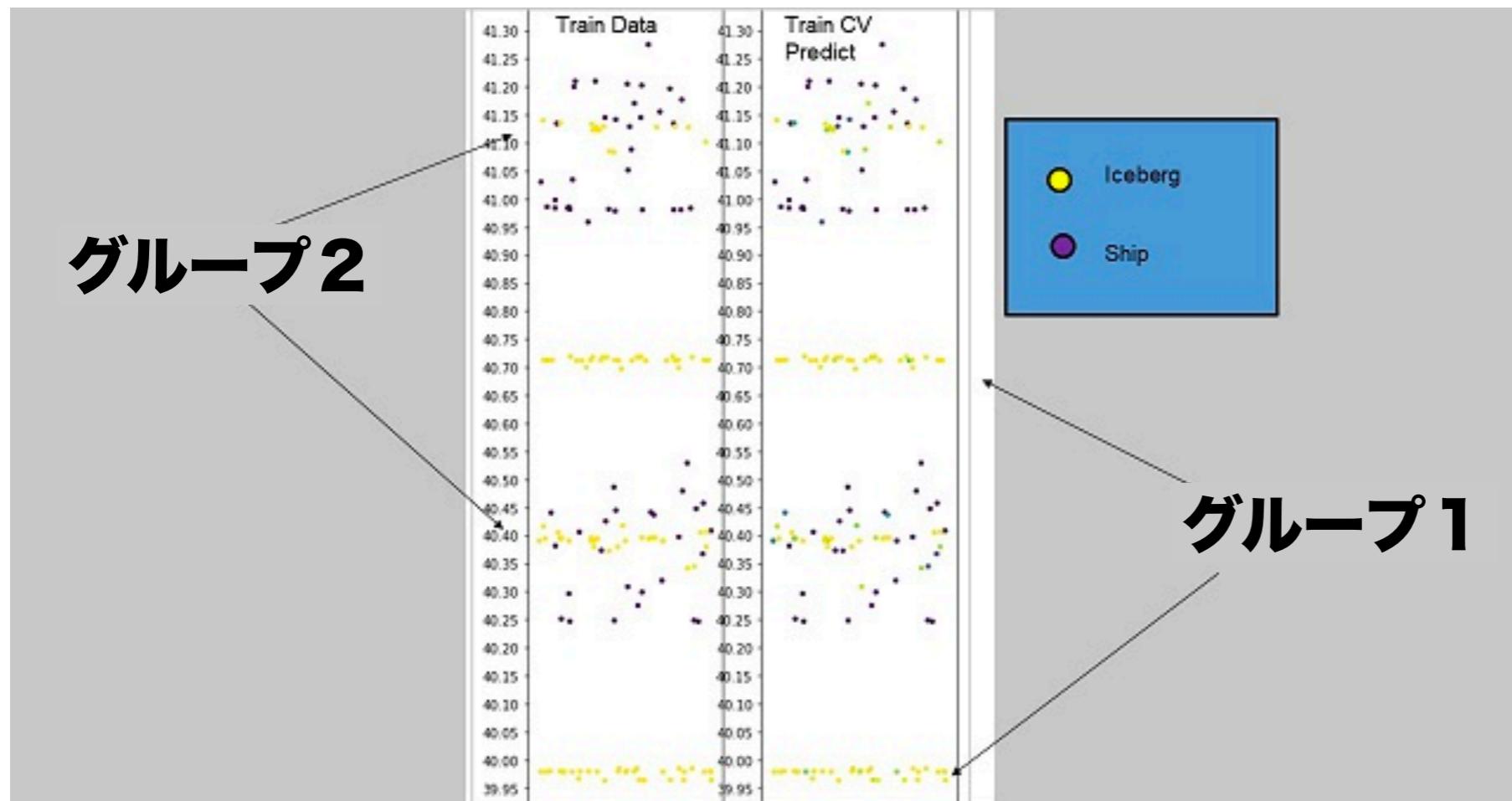


1. 5個の NN モデルを作成。5 fold CV でチューニング。
2. CV で学習したモデルの mean prediction を特徴量とする (量的変数)
3. 入射角でグループ化して集計 (mean, median, count) して特徴量作成
4. 入射角と (2) の mean prediction で KNN regressor (量的変数)
5. LGBM を (2, 3, 4) の特徴量で学習。合計で 5 つの特徴量。

# 1st: 入射角の適切なモデリング



入射角の可視化が鍵。> “Visualizing inc\_angle was the key finding that led us down our solution path.”



出典：<https://www.kaggle.com/c/statoil-iceberg-classifier-challenge/discussion/48241>

→ 入射角によって Unsupervised にグループ化してモデルを作る

# 再掲：どうやったら勝てるのか

大雑把に考えると今も昔も必要なことは変わらない。

EDA (探索的データ分析)

Validation (適切なモデル評価)

Survey (研究成果や過去解答から学ぶ)

# さいごに

- Kaggle ではアンサンブル以外が鍵となることがある。データからの気付き・NNのデザインなど。NN の発展や多様なデータのおかげ。
- 学習・議論・ゲームとしてなど様々な目的意識の参加者が共存している。興味を持っていただけだと Kaggle 愛好者として嬉しい。

# 付録：参考文献 (1/3)

1. [Blum & Hardt '15] "The Ladder: A Reliable Leaderboard for Machine Learning Competitions", In Proc. of the ICML '15. <https://arxiv.org/abs/1502.04585>
2. [Hardt '17] "Climbing a shaky ladder: Better adaptive risk estimation", <https://arxiv.org/abs/1706.02733>
3. [Rendle '10] "Factorization machines", In Proc. of the ICDM '10.
4. [Johnson & Zhang '11] "Learning Nonlinear Functions Using Regularized Greedy Forest", <https://arxiv.org/abs/1109.0887>
5. [Hu+ '17] “Squeeze-and-Excitation Networks”, In Proc. of the CVPR '17. <https://arxiv.org/abs/1709.01507>
6. [He+ '09] “Single Image Haze Removal”, In Proc. of the CVPR '09. <http://kaiminghe.com/cvpr09/>

# 付録：参考文献 (2/3)

7. [Zhang+ '17] "mixup: Beyond Empirical Risk Minimization", <https://arxiv.org/abs/1710.09412>
8. [Miech+ '17] "Learnable pooling with Context Gating for video classification", <https://arxiv.org/abs/1706.06905>
9. [Tian+ '16] "Detecting Text in Natural Image with Connectionist Text Proposal Network", In Proc. of ECCV '16 <https://arxiv.org/abs/1609.03605>
10. [Qi+ '17] "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation", In Proc. of the CVPR '17 <https://arxiv.org/abs/1612.00593>
11. [Ma+ '17] "Multi-View Deep Learning for Consistent Semantic Mapping with RGB-D Cameras", In Proc. of the IROS '17 <https://arxiv.org/abs/1703.08866>
12. [Shi+ '15] "An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition", <https://arxiv.org/abs/1507.05717>

# 付録：参考文献 (3/3)

13. [Arandjelović+ '16] "NetVLAD: CNN architecture for weakly supervised place recognition", In Proc. of the CVPR '16 <https://arxiv.org/abs/1511.07247>
14. [Zhou+ '17] "EAST: An Efficient and Accurate Scene Text Detector", In Proc. of the CVPR'17 <https://arxiv.org/abs/1704.03155v2>