



# 目次

- 自己紹介
- コンペ概要
- 解法紹介
- Appendix

※解法紹介中に登場する画像は実際の商標画像ではなく、 フリー素材を使用しています

# 自己紹介



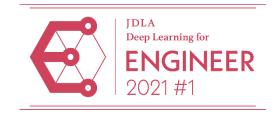
・名前: 穴井 晃太

・所属:トヨタテクニカルディベロップメント(株)

・ **> :** あにゃい (@rikunchu\_28)

・機械学習を活用した分析業務に従事

・一児のパパ





・参加理由:データ面白そう&弊社の特許チームへ取組み還元

# コンペ概要

## **Task**

Query画像に類似した画像をDatabaseから20件抽出(Image Retrievalタスク)

## **Evaluation**

Recall@20  $\Rightarrow$  ・各Queryの予測20件に正解が含まれていれば 1/ 含まれていなければ 0

・上記を全Queryで平均した数値

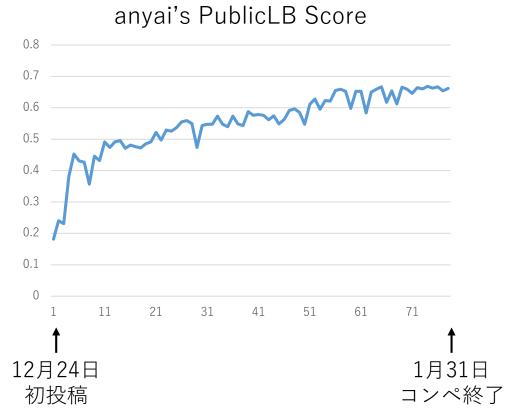
## **Data**



# コンペ概要

## **Result**

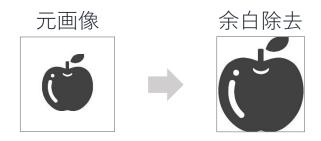
	順位		ユーザー名	最終スコア
<b>T</b>	1	The state of the s	tmsbir	0.734111
<b>T</b>	2		anyai	0.684824
		あなたの	カスコアは0.684824でした!お疲れさまでした!	
<b>T</b>	3	The state of the s	TDX	0.666666
	4		team FA	0.630350
	5	The state of the s	astamuse	0.618677



## **Preprocess**

・画像の余白除去

チュートリアルを参考に、二値化処理の閾値を調整して実施 (学習・推論時は周囲をpaddingして使用)

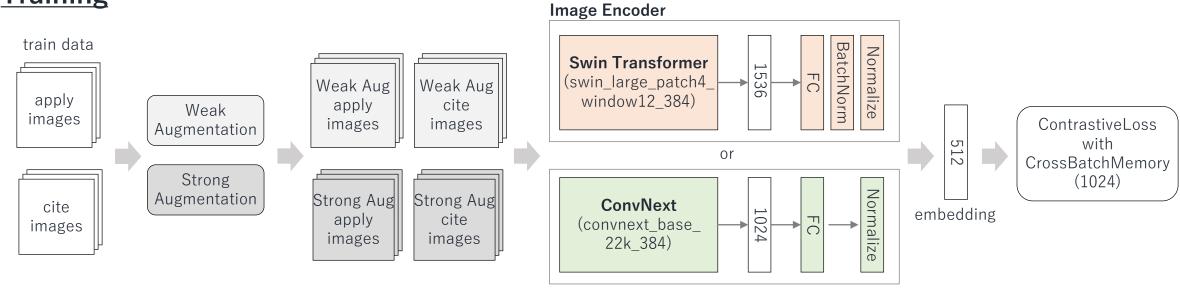


・Trainデータのグループ化

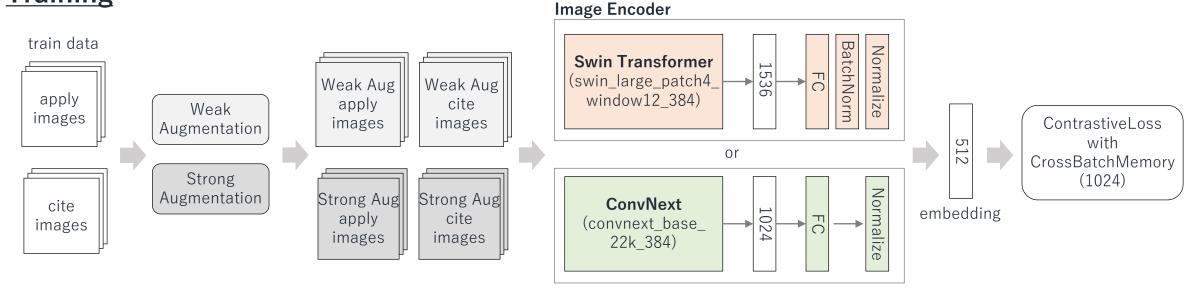
cite idが重複しているapply idを同一グループ化 (Metric Learning時のPositive/Negativeサンプリングノイズ低減)

apply id	cite id	category	Group
Α	В	-	1
С	D	-	2
Е	В	-	1

## **Training**



## **Training**



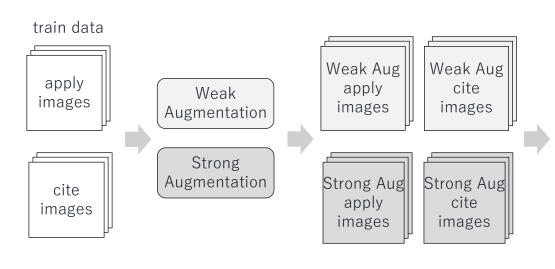
## **Weak Augmenation**

- · RGB Shft
- · Gray化
- Horizontal Flip
- · Random Resized Crop(0.7~)

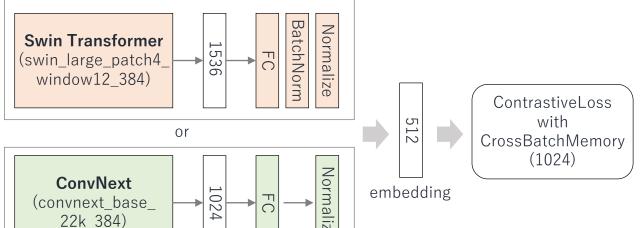
# 大画像 ・RGB Shft ・Gray化 ・Horizontal Flip ・Gaussian Blur ・Zoom Out (0.6~) ・Random Resized Crop(0.4~) or ・疑似ロゴ作成(Auglyを応用)

補足:Vertical Flip使用すると精度悪化

## **Training**







#### **ContrastiveLoss**

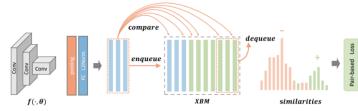
losses.ContrastiveLoss(pos\_margin=0, neg\_margin=1,

#### Equation:

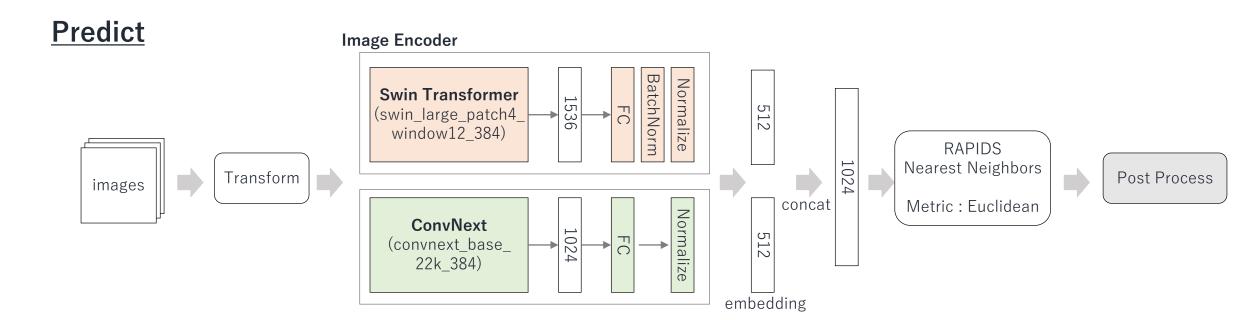
$$L_{contrastive} = [d_p - m_{pos}]_+ + [m_{neg} - d_n]_+$$

- ·Positive間はユークリッド距離 0
- ・Negative間はユークリッド距離 1 に近づくよう学習

### CrossBatchMemory



embedding結果を数batch分保持しておき、 pairwise lossの比較対象(Negativeサンプル)を増やす → batchサイズ小さくても学習安定



#### **Predict Image Encoder** BatchNorm Normalize **Swin Transformer** 1536 (swin\_large\_patch4\_ window12 384) **RAPIDS** Nearest Neighbors Transform Post Process images concat Metric: Euclidean

embedding

1024

Trainデータで誤判定となるデータを確認した結果、 queryに対し 正解データより類似したciteが多数存在する 傾向が見られた

ConvNext

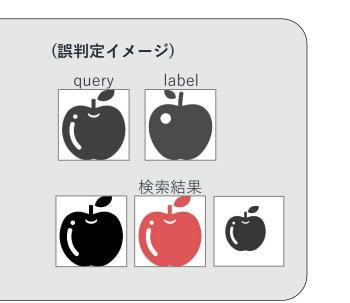
(convnext base 22k 384)



queryの近傍探索のみではスコア改善が困難であり、 **人間のアノテーション活用**が重要と判断

#### **Post Process**

- ・通常の近傍探索を上位15件 採用
- ・queryに似ているtrain apply画像を検索し、citeラベルを予測に追加 (+0.05~0.07 改善)
- ・ベクトルをAugmentation (query apply + cite) して再検索し、 結果を予測に追加 (+0.002~0.006 改善)



# **Appendix**

## **Training Tips**

- ・Swin Transformerのlearning rate
  Kaggleでは1e-5~1e-6が散見されるが 1e-4で大きめに設定すると学習良好
- ・CrossBatchMemory 256, 512, 1024, 2048で試して1024が良好。大きいほうが良いが、trainデータ数とのバランスも大事

## **Not Work**

- ・アノテーションされていないciteデータの活用
  - 強めのAugmentationで類似画像作成
  - 学習済みモデルでEmbedding作成し、類似データをグループ化(Pseudo Label)
- ・GeM Pooling なぜか使用すると学習がうまくいかず(おそらくP値をTrainableにしたため。今後は定数も試行する)
- ・OCRを使用した文字マスキング マスキング有無でロゴに着目した学習ができると期待したが、精度向上せず
- ・Google Landmarkコンペの優勝解法にあった DOLG

# **Appendix**

## **Loss Function**

Late Subをした結果、 1位チーム解法の通り ContrastiveLossよりもMultiSimilarityLossが良好。

ContrastiveLoss :全サンプルで一様に作用

・MultiSimilarityLoss: Hardサンプルを重視 ⇒ 類似データの多い本コンペに適していたと想定

#### MultiSimilarityLoss

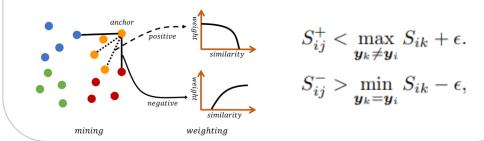
losses.MultiSimilarityLoss(alpha=2, beta=50, base=0.5, \*\*kwargs)

#### Equation:

$$\mathcal{L}_{MS} = \frac{1}{m} \sum_{i=1}^{m} \left\{ \frac{1}{\alpha} \log \left[ 1 + \sum_{k \in \mathcal{P}_i} e^{-\alpha(S_{ik} - \lambda)} \right] + \frac{1}{\beta} \log \left[ 1 + \sum_{k \in \mathcal{N}_i} e^{\beta(S_{ik} - \lambda)} \right] \right\}$$

#### 類似度が

- · max(Negative)+  $\varepsilon$  < Positive
- ・min (Positive)  $\varepsilon$  > Negative を重みづけしてLoss計算



# **Appendix**

## **References**

DeNA Engineering 『Facebook Al主催の画像のコピー検知のコンペで入賞した際の取り組み』 <a href="https://engineering.dena.com/blog/2021/12/fb-isc-1st/">https://engineering.dena.com/blog/2021/12/fb-isc-1st/</a>

⇒Trainingの流れやAugly応用など、めちゃくちゃ参考になりました

## Code

