

医療画像と 自己教師あり学習

大阪大学医学部 安部政俊

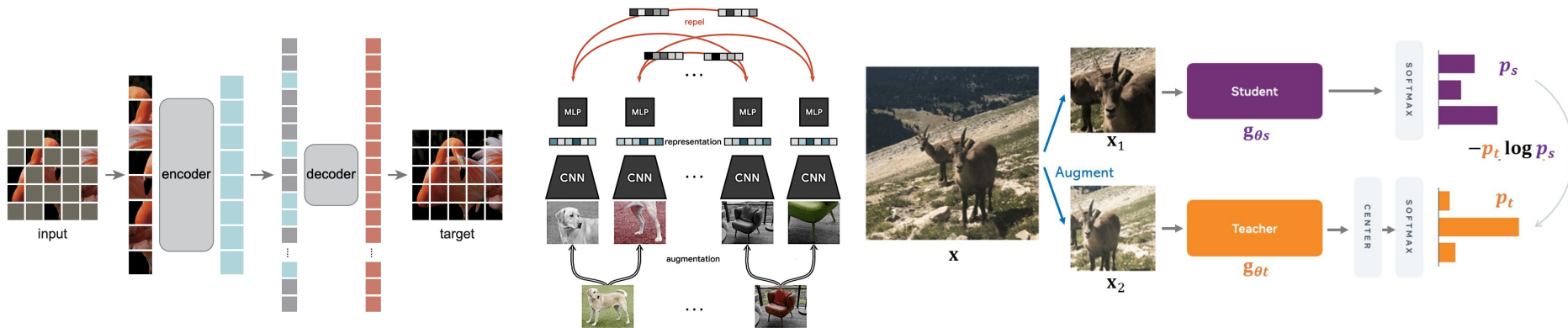
- 用語の説明
 - 自己教師あり学習とは
 - MILとは
 - WSIとは
 - 眼底画像とは
- 実例の紹介
 - SSiT
 - giga-SSL
 - ContIG

用語の説明

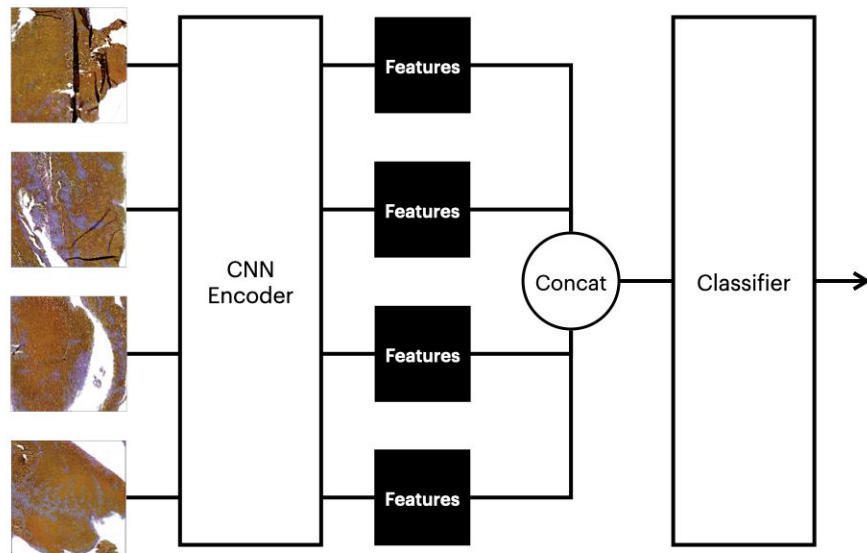
自己教師あり学習とは

ラベルなしデータに対して、データ自身から独自のラベルを機械的に作成したもののから画像の表現を獲得するタスク

- 画像の一部をmaskして隠された部分を再構成できるように学習する (e.g:MAE)
- 画像A(正例)とそれにaugmentationを加えた画像A'、別画像B(負例)を用意して、それぞれを埋め込んだときに似ているAとA'が近く、異なるBは遠くなるように学習する (e.g:SimCLR)
- 画像Aに異なる摂動を加えたA'とA''を用意して別々に埋め込みを得た時にA'とA''が似ていることを学習する(e.g:BYOL,DINO)



MultInstanceLearning(MIL)とは

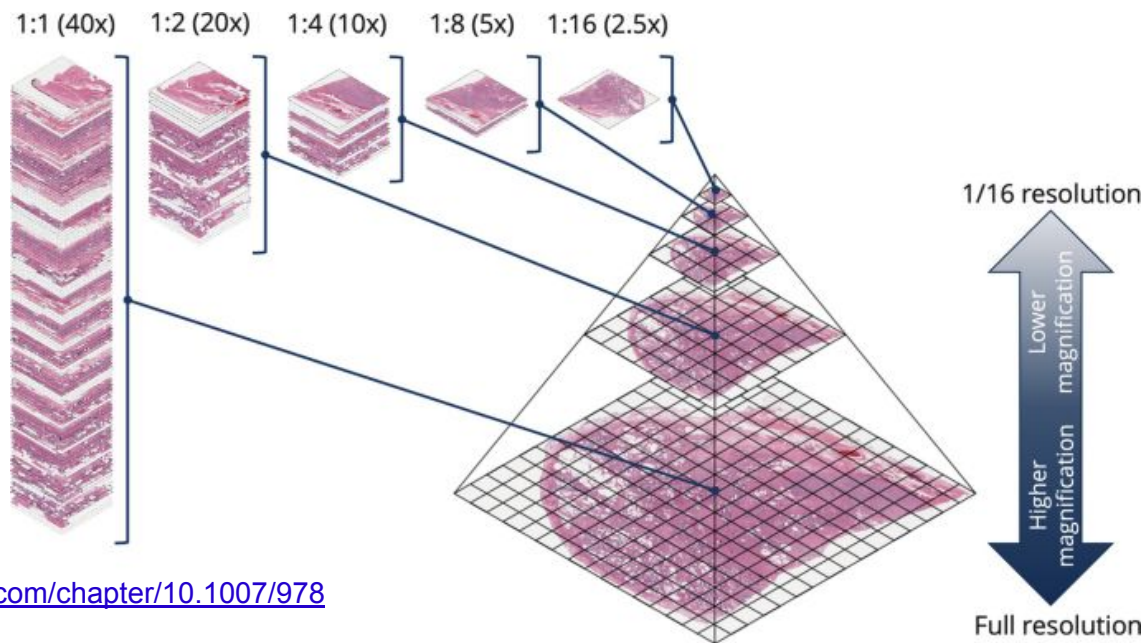


- 巨大画像をpatchに分割した後など、“複数の画像をbackboneに入力したい”ときに使える
- 各々の画像をbackboneに入れて埋め込み獲得
- 埋め込みを集約して最終のMLPにいれる
- kaggleではPANDA,OSIC,RSNA2022c-spineなどでみられた
- いろんな論文が出ている集約層の工夫は kaggleではそんなに有効ではないイメージ

Whole Slide Imaging(WSI)とは

Whole slide images (WSI)とは、バーチャルスライドとも呼ばれる病理組織プレパラート標本全体をスキャンしたもの。

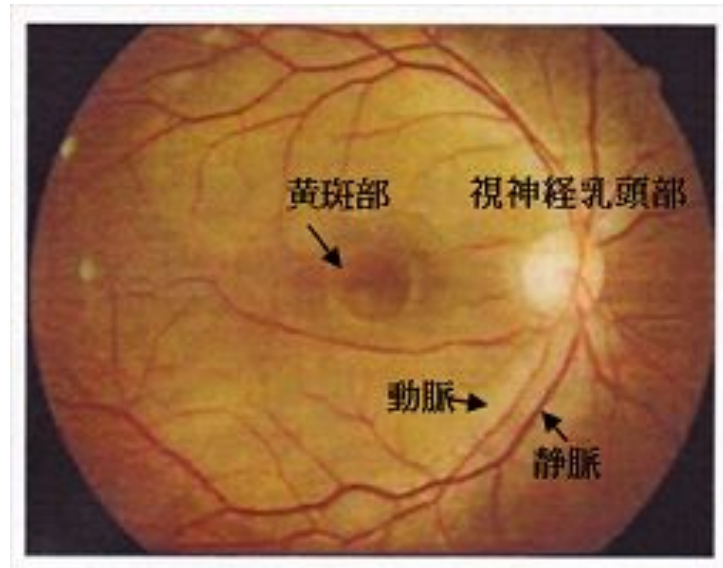
画像サイズが巨大なので128*128,256*256などのパッチに分割して扱うことが多い。パッチの作成時に組織以外の余白を除くことが多い。



眼底画像とは

眼底の血管、網膜、視神経を撮影した写真

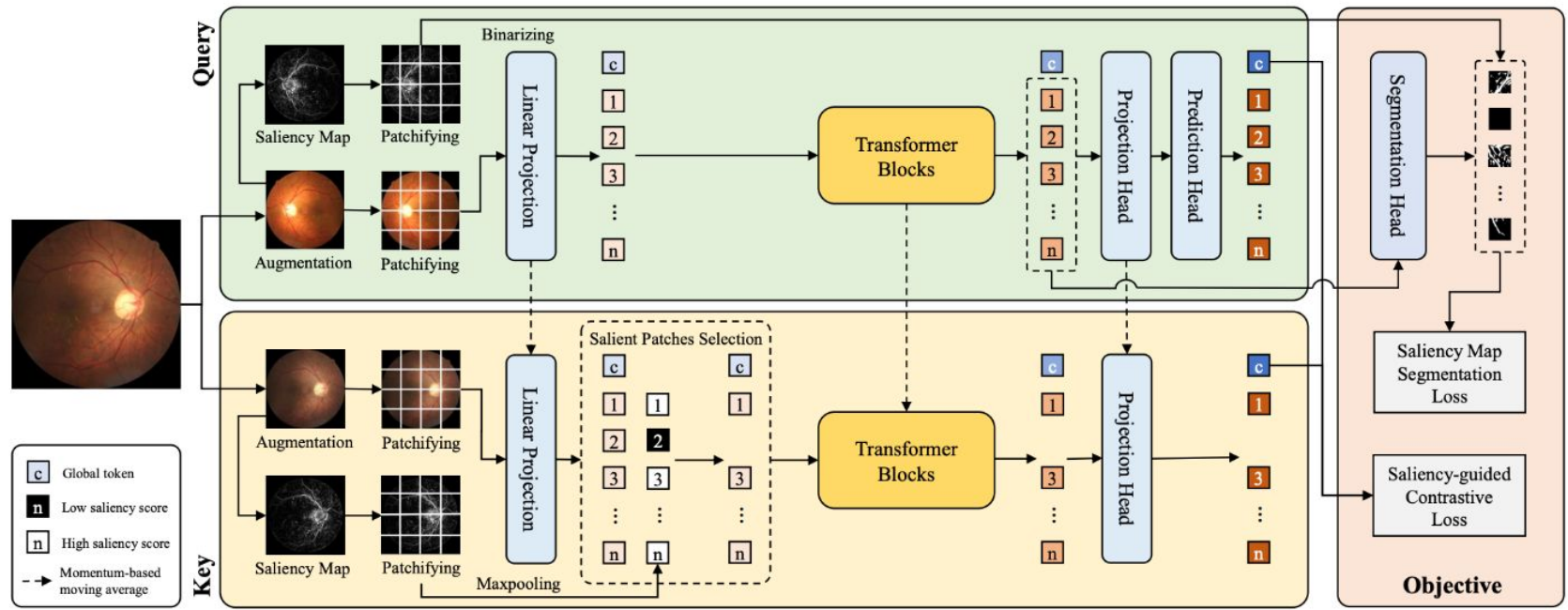
血管の様子をみることで高血圧や糖尿病、動脈硬化を、視神経乳頭をみることで緑内障を診ることができる。



https://nerima-hosp.or.jp/about_us/hospital/column/kensa/01-2/#:~:text=%E7%9C%BC%E5%BA%95%E5%86%99%E7%9C%9F%E3%81%AF%E4%BA%BA%E4%BD%93%E3%81%AE,%E3%81%AE%E7%AA%93%E3%80%8D%E3%81%A8%E3%82%82%E8%A8%80%E3%81%88%E3%81%BE%E3%81%99%E3%80%82

実例の紹介

SSiT 負例あり+再構成



- 一つの画像に"Key", "Value"用に別のaugmentationをかけてpatchに埋め込む
- Saliency Mapを画像処理で作成してMapが黒い部分のpatchは"Key"側ではbackboneの入力から省く
- patch埋め込みをbackboneのViTに入力→headに通して"Key", "Value"の出力を得る
- この埋め込みの類似度が高くなるよう / 別画像の埋め込みとは類似度低くなるよう学習 + Saliency Mapのsegmentationも学習

<https://arxiv.org/pdf/2210.10969v1.pdf>

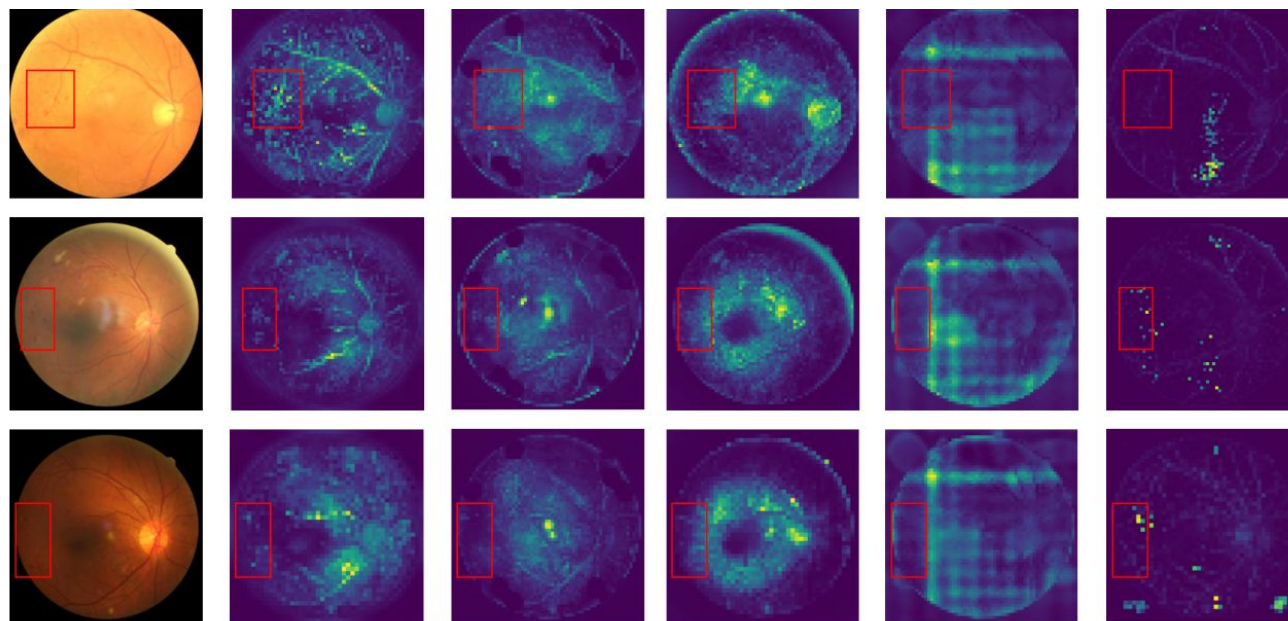
TABLE I

COMPARISON RESULTS WITH STATE-OF-THE-ARE SELF-SUPERVISED LEARNING METHODS ON THREE EVALUATION DATASETS.

Method	DDR			Messidor-2			APTOS2019		
	Fine-tuning	Linear	k -NN	Fine-tuning	Linear	k -NN	Fine-tuning	Linear	k -NN
Random init.	18.40 \pm 7.09	15.56 \pm 5.60	19.42	19.55 \pm 2.40	15.80 \pm 7.14	5.06	70.87 \pm 0.59	57.99 \pm 2.19	56.48
ImageNet init.	84.27 \pm 0.74	67.06 \pm 0.75	40.47	78.90 \pm 1.49	61.14 \pm 1.63	33.54	93.37 \pm 0.45	86.59 \pm 0.43	74.71
SimCLR [12]	72.15 \pm 0.83	67.69 \pm 1.68	53.53	65.47 \pm 1.45	59.28 \pm 2.02	40.87	90.34 \pm 0.29	86.79 \pm 0.60	83.29
MoCo-v3 [14]	71.41 \pm 1.61	66.08 \pm 1.15	56.05	63.46 \pm 0.74	59.82 \pm 2.05	40.41	90.15 \pm 0.58	85.19 \pm 0.53	80.68
DINO [28]	68.49 \pm 0.94	66.38 \pm 0.53	49.86	60.47 \pm 2.39	60.98 \pm 0.92	40.65	90.50 \pm 0.32	85.07 \pm 0.37	79.44
MAE [31]	72.40 \pm 1.91	59.31 \pm 0.89	38.70	51.35 \pm 4.28	43.50 \pm 1.59	21.39	90.97 \pm 0.35	83.24 \pm 0.54	75.64
SSiT (ours)	81.88 \pm 0.26	71.33 \pm 0.78	58.89	77.53 \pm 0.84	67.23 \pm 0.53	49.42	92.97 \pm 0.29	89.65 \pm 0.20	84.65

- fine tuning時はMoCov3以外事前学習時より高解像度 (384*384)を使用
- 対照学習系は300epoch,再構成系は800epoch学習
- 線形層でもfine tuningでも従来手法よりも性能が良いらしい
- APTOSの結果はkaggleのtestsetは使わずtrainingを自分で分割して計算していることに注意

SSiT 可視化



Fundus images

SSiT (ours)

DINO

SimCLR

MoCo-v3†

ImageNet init.†

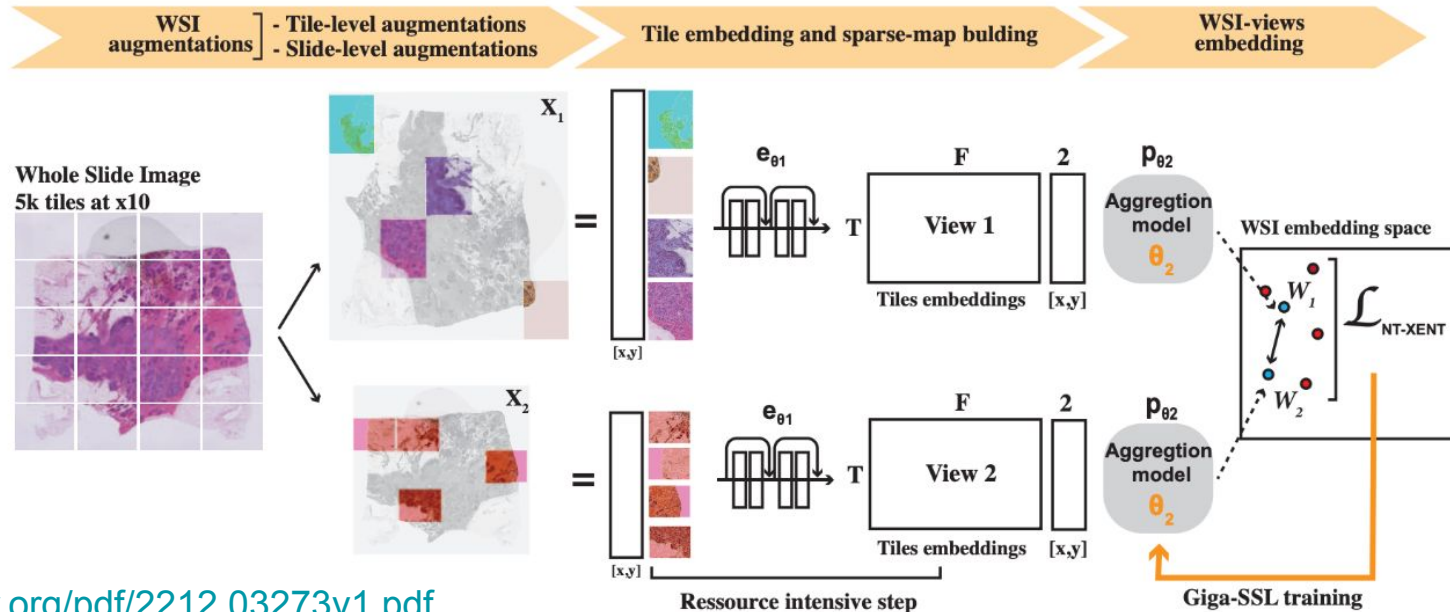
入力は1024*1024

SSiTとDINOのアテンションマップのみ、眼底画像のシーンレイアウトを表している。しかし、DINOはDR関連病変を見落とす可能性が高く、SSiTは対応する診断領域をより正確に、より明確な境界で強調する(図 3の赤枠部分)。一方、SimCLRは明るい領域に焦点を当て、意味的な情報を識別していない。MoCo-v3やImageNet初期化ViTは、意味のある領域を表示することすらできません。

らしい

- MoCo v3をベースにsaliency mapの再構成や saliency mapを用いたmaskingを使って従来手法より性能向上
- 再構成を学習しているのでより細かい情報が抽出できそう
- saliency mapの計算は関心オブジェクトと背景の間に明確な強度差が存在するOCT、MRI、胸部X線などの他の種類の医療画像にも適用可能

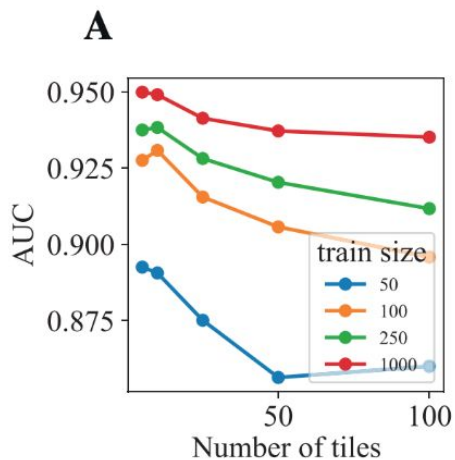
giga-SSL SparseConvMILのtile埋め込み/集約層を対照学習



<https://arxiv.org/pdf/2212.03273v1.pdf>

- tileレベルでmoco-resnetをあらかじめ学習
- tile埋め込み器をfreeze、ランダムにtileをK枚ずつ選んでresnetで埋め込み
- タイルが元々合った場所に埋め込みをマッピングした“スパースマップ”を作成
→augmentationかける
- スパースマップを畳み込んで集約

giga-SSL 性能



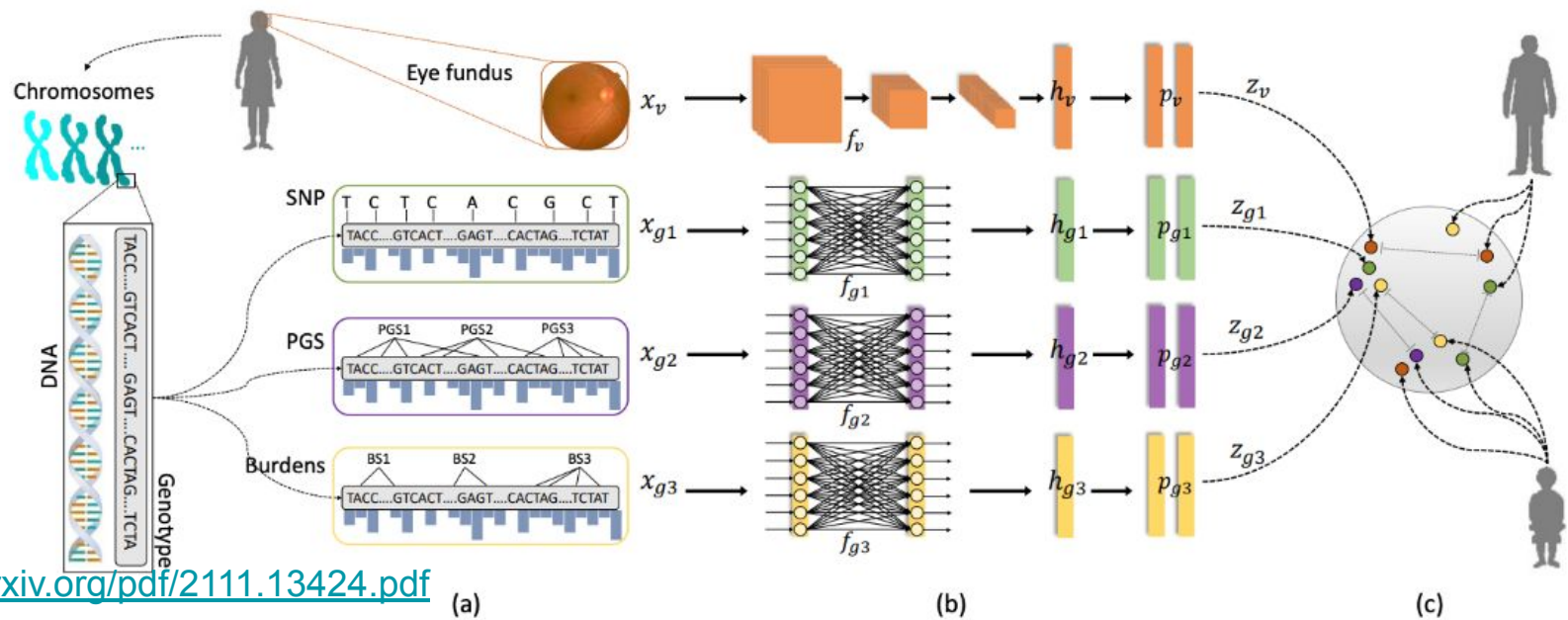
Task	Method	Giga-SSL (proposed)	AverageMIL	DeepMIL [17]	HIPT [3]	DeepSMILE [30]
	Linear	✓	✓	✗	✗	✗
Task	% data					
NSCLC _{subtyping}	100	0.952 ± 0.020	0.913 ± 0.023	0.948 ± 0.017	0.952 ± 0.021	-
	25	0.939 ± 0.017	0.885 ± 0.036	0.922 ± 0.034	0.923 ± 0.020	-
BRCA _{subtyping}	100	0.905 ± 0.032	0.859 ± 0.038	0.874 ± 0.050	0.874 ± 0.060	-
	25	0.890 ± 0.058	0.822 ± 0.072	0.860 ± 0.042	0.821 ± 0.069	-
RCC _{subtyping}	100	0.982 ± 0.007	0.973 ± 0.011	0.986 ± 0.008	0.980 ± 0.013	-
	25	0.975 ± 0.012	0.959 ± 0.015	0.970 ± 0.016	0.974 ± 0.012	-
BRCA _{molecular}	100	0.938 ± 0.035	0.920 ± 0.037	0.924 ± 0.042	-	-
	25	0.853 ± 0.075	0.799 ± 0.068	0.810 ± 0.093	-	-
BRCA mHRD	100	0.756 ± 0.028	0.706 ± 0.030	0.736 ± 0.047	-	0.727 ± 0.010
	25	0.743 ± 0.039	0.643 ± 0.050	0.660 ± 0.046	-	-
BRCA tHRD	100	0.855 ± 0.023	0.799 ± 0.034	0.836 ± 0.052	-	0.838 ± 0.012
	25	0.781 ± 0.050	0.698 ± 0.078	0.721 ± 0.075	-	-

用いるタイル枚数は少ない方が良い
推論時にタイル枚数を変えるのは良くない

従来手法よりもよい or 同等
集約層を畳み込みから attention に変えると性能悪化

- MILの従来手法はタイル同士の空間的な依存性を考慮していないものが多い(attentionで集約するとタイルが元々のスライドのどこにあったのかの情報は失われてしまう)
- SparseConvMILならスパースマップ作成によりタイル間の空間依存性を考慮
- SparseConvMILに自己教師あり学習を組み合わせて性能向上

ContIG SIMCLRを遺伝子データでもやる



<https://arxiv.org/pdf/2111.13424.pdf>

- 画像をresnetで、遺伝子データをMLPで埋め込んで対照学習のloss計算(SIMCLRと同じloss)
- $loss_total = loss_image * alpha + loss_gene * (1 - alpha)$ で計算(<https://github.com/HealthML/ContIG>)
- 画像埋め込みと遺伝子埋め込みの内積をとっているわけではないのでCLIPlikeではない
- PGS: 多くの主に一般的なSNPを集約→特定の疾患に対する人の遺伝的感受性を反映する単一のスコア
個々のSNPは、疾患との関連性の強さに基づいて重み付けされる。異なる形質や疾患に対して多くの異なるPGSを使用することで、個人の複雑な形質素因を多面的にみる
- Burdensスコア: 局所的な遺伝子領域内の複数の希少バリエントを集約したもの

ContIG 性能

Model & Genetics Encoder		APTOS	RFMiD	PALM	Cardio. Risk Pred.	
		QwKappa ↑	ROC-AUC ↑	Dice-Score ↑	MSE ↓	ROC-AUC ↑
Baseline	-	80.47	91.64	77.25	3.440	56.29
SimCLR [23]	-	81.83	91.88	70.41	3.451	59.38
SimSiam [25]	-	75.44	91.28	72.26	3.442	57.37
BYOL [40]	-	71.09	89.88	66.32	3.414	59.73
Barlow Twins [130]	-	72.28	92.03	70.53	3.430	59.05
NNCLR [32]	-	77.93	91.89	72.06	3.426	61.95
ContIG (Raw-SNP)	H1	84.01	93.22	76.98	3.254	70.10
ContIG (PGS)	H1	85.93	<u>93.31</u>	78.47	<u>3.176</u>	72.72
ContIG (Burden)	H1	83.22	93.03	76.49	3.160	<u>72.37</u>
ContIG (Inner RPB)	H1	81.52	92.95	<u>77.34</u>	3.202	70.80
ContIG (Outer RPB)	H1	<u>84.22</u>	93.62	<u>76.97</u>	3.187	71.80

Table 1. Downstream evaluation results by fine-tuning on each task. **Bold** indicates the best result, underlined is second best. RPB in our method stand for the genetic modalities used: Raw-SNPs, PGS-scores, and Burden-scores. ↑ means higher is better, and ↓ lower is better.

遺伝子データ使った方が良かった？

APTOSのtestはkaggleのtrainの一部を使用している