# RankMix: Data Augmentation for Weakly Supervised Learning of Classifying Whole Slide Images with Diverse Sizes and Imbalanced Categories

Yuan-Chih Chen, Chun-Shien Lu
IIS, Academia Sinica, Taiwan, ROC
THU-PM-318

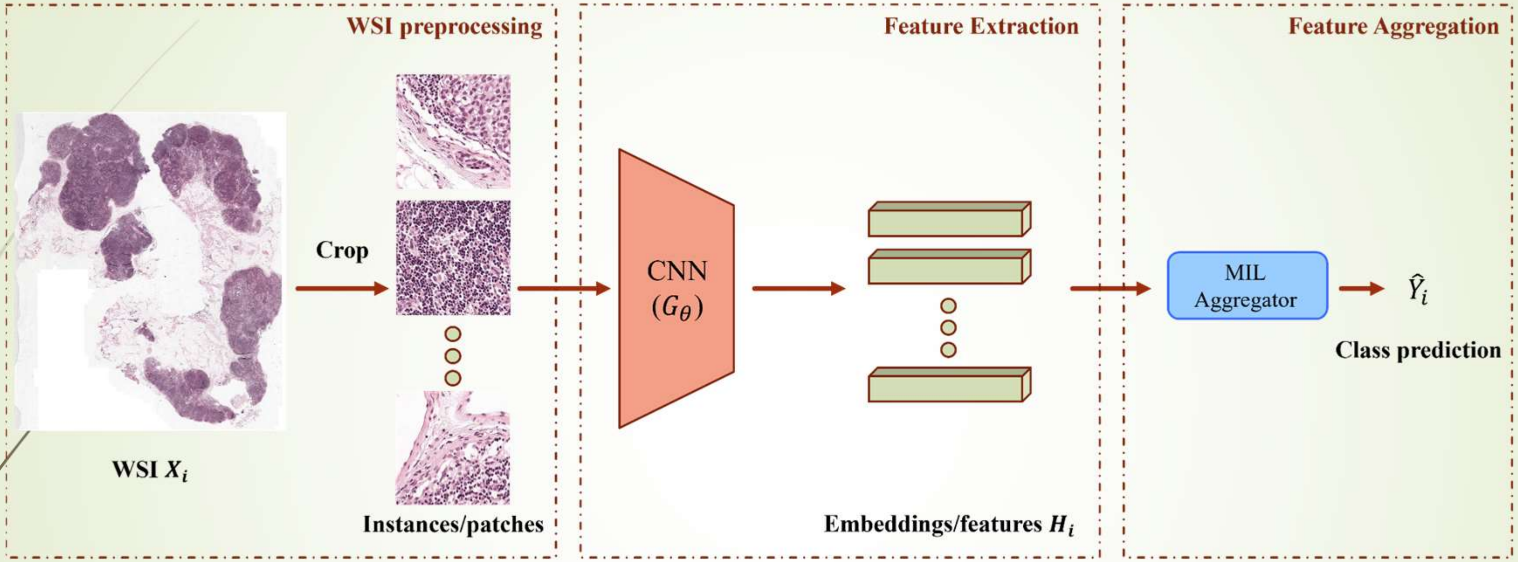
中央研究院 資訊科學研究所
Institute of Information Science, Academia Sinica
SINCE 1982

JUNE 18-22, 2023
CVPR
VANCOUVER, CANADA

# Preview

- Whole Slide Images (WSIs) are usually gigapixel in size and lack pixel-level annotations.
- In this study, we propose, RankMix, a data augmentation method of mixing ranked features in a pair of WSIs to improve the performance of WSI classification.
- RankMix introduces the concepts of pseudo labeling and ranking in order to extract key WSI regions in contributing to the WSI classification task.
- A two-stage training is further proposed to boost stable training and model performance

# Whole Slide Image (WSI) classification




- A slide will be cropped into tens of thousands of $224 \times 224$ patches and ignore the background patches.
- Then, the embeddings of patches will be fed into feature aggregator to get the overall class prediction.

# Motivation

- We only have a slide-level label without any information about patches.
  - tens of thousands of 224×224 patches mapping to one slide label.
- WSI datasets often only have 100-1000 slides and may have the problem of class imbalance due to rare diseases.

| | Camelyon16 | TCGA-Lung | WSI-usability |
|---|---|---|---|
| Class1 (slides) | 160 | 512 | 23 |
| Class2 (slides) | 240 | 534 | 427 |
| Total (slides) | 400 | 1046 | 450 |

Lack of data and class imbalance

use Mixup to increase training samples and mitigate the class imbalance problem

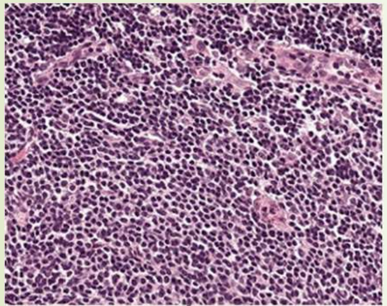
# Challenge for Mixing Two WSIs

- Two WSIs may be hundreds of times the size of the other.

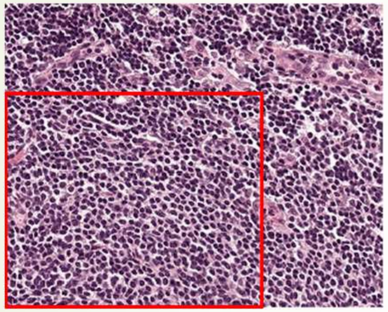
| | Camelyon16 | TCGA-Lung | WSI-usability |
|---|---|---|---|
| Maximum number of patches of a WSI | 44000 | 12700 | 120000 |
| Minimum number of patches of a WSI | 1200 | 50 | 700 |
| The ratio of maximum patches and minimum patches | 36.66 | 254 | 171.42 |

- Cannot simply resize patches due to the loss of background patches and the necessary of remaining the same scanning magnitude.
- Cannot use Cutout techniques due to the tumor may only occupy small region. (In Camelyon16, the tumor area only accounts for approximately less than 10% of the tissue area in the positive slide.)

# How to deal with the large gap between two slides of different size?

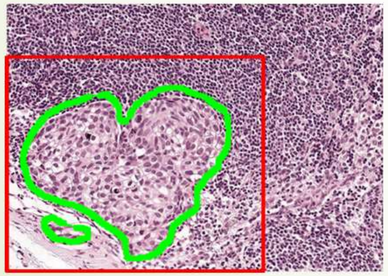


Normal slide



The part region of normal slide

Tumor slide



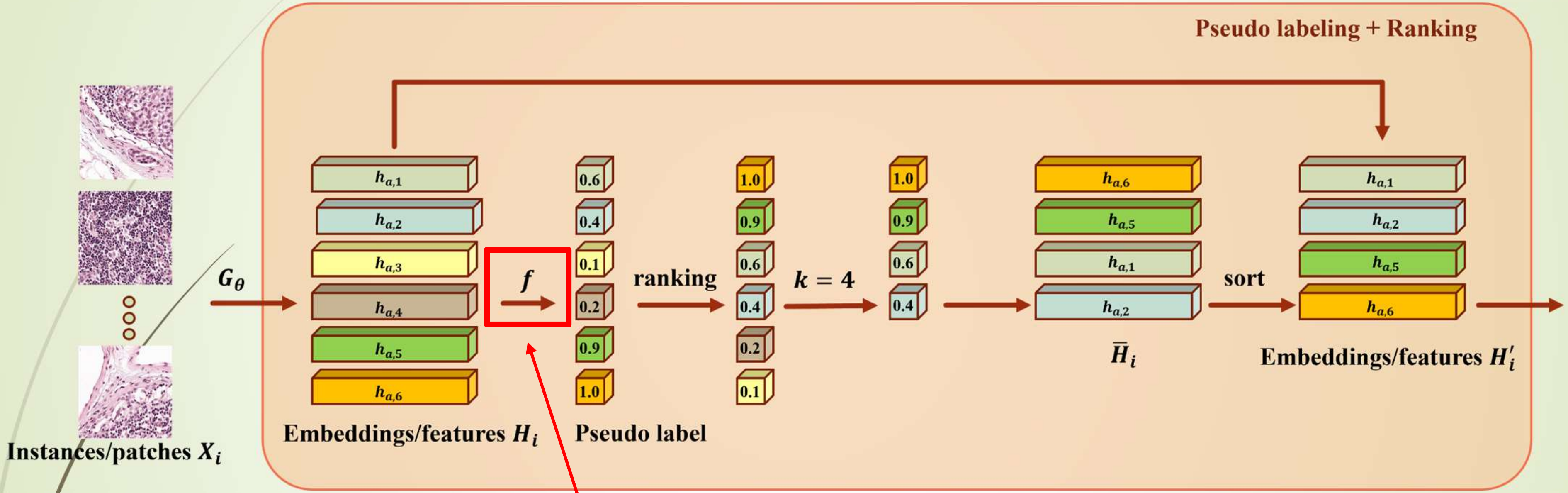
The part region of tumor slide

- We can’t resize the slide to match the other one due to:
  - The loss of background patches.
  - The large gap between two slides of different size.

- If we can find the red region as shown in the left figure, we can mix these two regions very easily.

However, we only have a slide-level label without any information about patches.

# How to get the red box region?




- If we have a score function $f$ to predict the class probability (pseudo label) of a patch, we can get the red box region mentioned in the previous page very easily.
- We can get the arbitrary number $k$ of patches as shown in figure ($k = 4$ for the illustration)

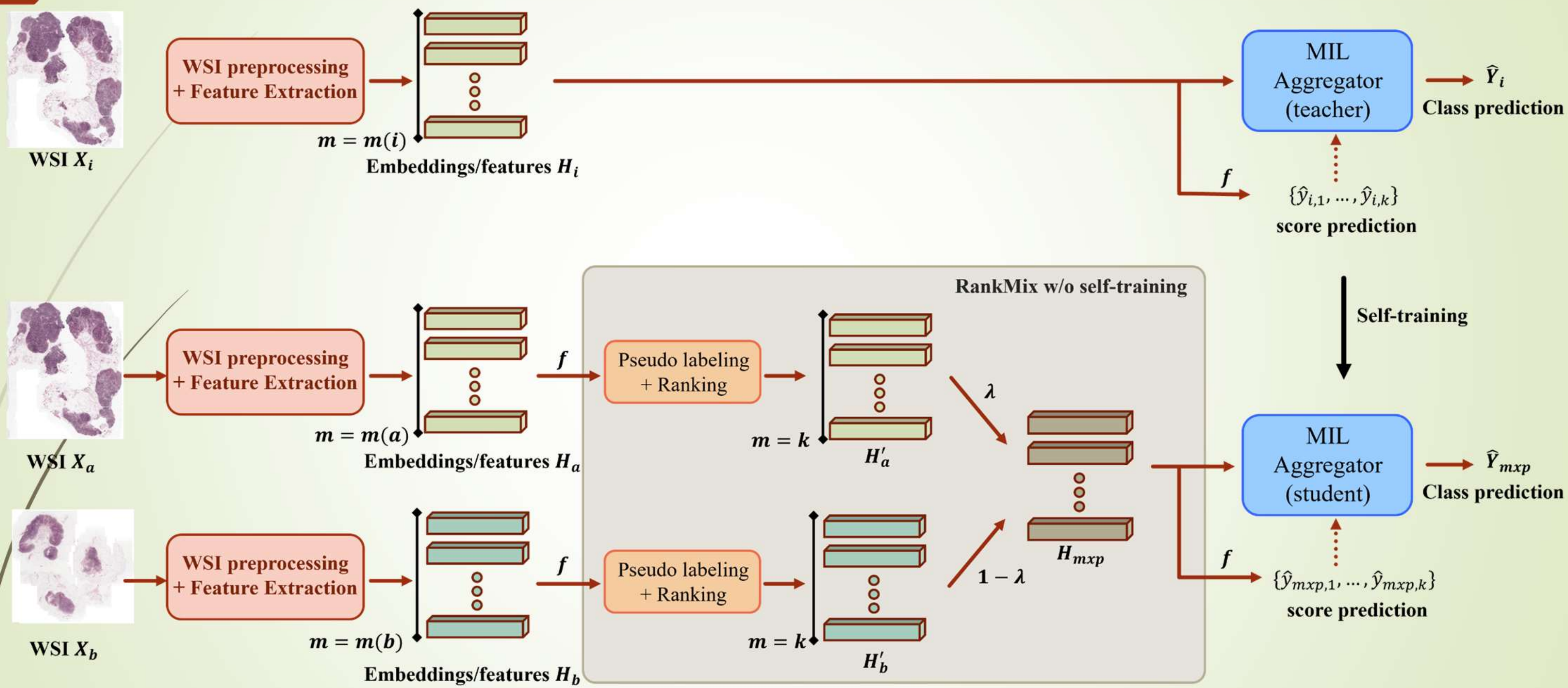
# How to get the score function $f$ ?

- In multiple instance learning (MIL), the common approach is to extract the feature embedding of patches of a slide then make a decision based on these patch embeddings such as:
  - Instance-based approach
  - Attention-based approach
  - Clustering-based approach
- The score function $f$ can be:
  - The instance classifier for instance-based approach
  - The attention weight for attention-based approach
  - The distance to cluster center for clustering-based approach
- Because the existing approaches are predicting the overall class based on all patches of a slide, we can often find the similar mechanism in existing methods.

The proposed method can be applied to the most existing MIL approaches.

# Self-training

How can we get better score function $f$ ?



# Self-training

How can we get better score function $f$ ?

- In the first stage, we train the aggregator by general MIL approach without mixup samples.
  - We can get decent performance of MIL aggregator as proved in previous general MIL works.
  - A easier task (compared to mixup samples) may avoid unstable training.
- In the second stage, we train the MIL model (student model) with mixed samples and utilize the model from the first stage (teacher model) to make pseudo labels (the concept of self-training).

| Method/Dataset | Camelyon16 | | | WSI-usability | | | TCGA-Lung | | |
|---|---|---|---|---|---|---|---|---|---|
| | ACC | AUC | AUPRC | ACC | AUC | AUPRC | ACC | AUC | AUPRC |
| DSMIL [28] | 86.82% | 93.32% | 92.68% | 76.11% | 86.60% | 24.51% | 93.81% | 97.89% | 97.75% |
| + ReMix [46] | 82.17% | 86.89% | 83.86% | 83.19% | 85.83% | 25.59% | 94.29% | 97.62% | 97.29% |
| + RankMix w/o self-training | 87.60% | 92.07% | 92.43% | 90.27% | 87.07% | 25.66% | 94.29% | 98.00% | 97.76% |
| **+ RankMix** | **89.92%** | **93.47%** | **92.74%** | **90.27%** | **88.16%** | **28.41%** | **94.29%** | **98.04%** | **97.79%** |
| FRMIL [10] | 89.15% | 94.57% | 93.66% | 83.19% | 87.69% | 45.99% | 90.95% | 95.38% | 94.96% |
| + ReMix [46] | 82.59% | 87.29% | 87.35% | 89.25% | 80.63% | 33.09% | 92.22% | 96.99% | 97.04% |
| + RankMix w/o self-training | 90.70% | 94.11% | 93.68% | 80.53% | 84.27% | 38.55% | 93.33% | 95.84% | 97.01% |
| **+ RankMix** | **91.47%** | **94.59%** | **93.99%** | **93.81%** | **93.61%** | **47.65%** | **93.33%** | **97.00%** | **97.04%** |

# How to train the student model

- If the student model is the same as the teacher model:
  - The teacher model fixed
  - The teacher model changed:
    - Fine-tuning approach (just like BERT-based method)
- If the student model is different from the teacher model
  - Knowledge distillation approach
- In our experiment, we find that the fine-tuning method has the best performance.

Any knowledge transfer methods will be useful

# Conclusion

- How do we get the smaller slide but it can still remain significant?
  - Use the conception of self-train and pseudo labeling.
  - May remove some noise from patches.
  - We can get a smaller slide which can represent the original one

- Why we need to use mixup?
  - Mixup has the chance to improve the performance of model when suffer from the class imbalance (rare disease, etc.)
  - Mixup has the chance to improve the performance of model when the training data is scarce (expensive to collect data)
  - There are many mixup-based methods in natural image, we want to make these approach available for WSIs.