DLMI Final Project

ID: r12922075

Name: 吳韋論

Data

COVID-19 Chest X-Ray Database - Kaggle

Introduction

A team of researchers with medical professionals creates a dataset of chest X-ray images. The dataset includes images and corresponding lung masks for COVID-19, Lung Opacity (non-COVID lung infections), Normal, and Viral Pneumonia cases. The dataset aims to facilitate the classification of these disease categories and accurate segmentation of lung masks, aiding in enhanced diagnostic capabilities.

Dataset

The dataset comprises 21165 images and masks, each with a resolution of 300×300 pixels in PNG format. The images are classified into four categories.

- COVID-19: 3616 images, masks
- Lung Opacity: 6012 images, masks
- Normal: 10192 images, masks
- Viral Pneumonia: 1345 images, masks

Project Objective

Classification

- Try different **supervised, SOTA self-supervised, zero shot** classification methods
- Metric comparison
    - F1 score, precision, recall, accuracy
    - Training speed
    - Training dataset size
- Objective: Identify the most effective strategies for classifying chest X-ray images, enhancing diagnostic accuracy and efficiency.

Segmentation

- Try different **supervised** segmentation methods
- Metric comparison
    - Dice score
- Objective: Identify the most highest dice score method

Muti-Task Learning for Classification and Segmentation

- **Modify Unet Architecture** to enhance the encoder's output features for additional classification tasks.
- Objective: Evaluate the performance when simultaneously conducting classification and segmentation tasks.

Methodology

Part1. Classification

Supervised

I select six common models from torchvision, and all training data to fine-tune model. Below are their architectures and key features.

1. **Swin Transformer**
    - **Architecture**: Utilizes shifted windows for cross-window connection.
    - **Key Features**: Scalable, efficient, maintains long-range interaction processing.
2. **VIT Base**
    - **Architecture**: Applies transformer principles to image patches.
    - **Key Features**: Global image context understanding through sequential patch processing.
3. **ConvNeXt**
    - **Architecture**: Modernized ConvNet design inspired by transformers.
    - **Key Features**: Improved scalability and efficiency.
4. **DenseNet**
    - **Architecture**: Densely connected layers in a feed-forward fashion.
    - **Key Features**: Enhanced feature propagation, reduced parameters, feature reuse.
5. **EfficientNetV2**
    - **Architecture**: Optimization of EfficientNet with scaling for speed and efficiency.
    - **Key Features**: Compound scaling of network dimensions.
6. **ResNeXt101**
    - **Architecture**: Parallel residual transformations extend ResNet.
    - **Key Features**: Balances network depth and width for improved performance.

Self-Supervised (SOTA Method)

I chose two state-of-the-art self-supervised methods, freezing most layers from the original pretrained weights and **only fine-tuning the last few layers and the classification layer**. This approach aims to verify that it can accelerate training speed without significantly decreasing performance. Additionally, I observed the effects of using only 25%, 50%, 75%, and 100% of the training data.

1. **DINOv2 (Apr 2023 Meta AI Research)**
    - **Architecture**: Vision transformer using self-distillation without labels.
    - **Key Features**: Invariance to data augmentations, strong feature extraction.

2. **BEITv2 (Oct 2022 Microsoft Research)**
    - **Architecture**: Masked image modeling for pre-training transformers.
    - **Key Features**: Efficient unsupervised data leverage, improves with fine-tuning.

Zero-Shot

I use the CLIP model with prompts to test the zero-shot capabilities. The prompts are as follows:

- COVID-19: A chest X-ray image showing features characteristic of COVID-19, such as bilateral ground-glass opacities.
- Lung Opacity: A chest X-ray image showing lung opacity which might indicate conditions like pneumonia or edema but not specific to COVID-19.
- Normal: A chest X-ray image of normal lungs without any signs of infection, opacity, or other abnormalities
- Viral Pneumonia: A chest X-ray image showing signs of viral pneumonia, distinct from bacterial causes, possibly including patterns like patchy airspace opacities.

1. **CLIP: VIT Large**
    - **Architecture**: Combines visual and linguistic understanding.
    - **Key Features**: Classifies images using natural language, requires no fine-tuning.

Part2. Segmentation

Supervised

I choose 8 different types of popular models: Unet, Unet++, MAnet, FPN, PSPNet, PAN. DeepLabV3, and DeepLabV3+. Each model has unique characteristics explained below.

1. **Unet**
    - **Architecture**: Features a contracting path and a symmetric expanding path.
    - **Key Features**: Employs skip connections to enhance information flow and segmentation precision.
2. **Unet++**
    - **Architecture**: Builds on Unet with nested, dense skip pathways.
    - **Key Features**: Bridges the gap between encoder and decoder, enhancing detail capture.
3. **MAnet (Multi-Scale Attention Network)**
    - **Architecture**: Incorporates attention mechanisms into the segmentation framework.
    - **Key Features**: Uses multiple attention modules to enhance focus on important image regions.
4. **FPN (Feature Pyramid Network)**
    - **Architecture**: Creates a multi-scale feature pyramid.
    - **Key Features**: Utilizes features at different scales for improved object detection and segmentation.
5. **PSPNet (Pyramid Scene Parsing Network)**
    - **Architecture**: Employs a pyramid pooling module for global context.
    - **Key Features**: Handles complex scenes by capturing context at multiple levels.
6. **PAN (Pyramid Attention Network)**

- **Architecture**: Merges feature fusion and attention mechanisms.
- **Key Features**: Focuses on fine details across scales for better segmentation.

7. / 8. **DeepLabV3/DeepLabV3+**

- **Architecture**: Utilizes atrous convolutions and an encoder-decoder setup.
- **Key Features**: Captures detailed multi-scale information and refines segmentation edges.

Part3. Muti-task for Classification and Segmentation

Supervised

I modify Unet architecture to include a new branch after the encoder. This branch is specifically designed to perform classification tasks, utilizing the features generated by the encoder.

Multi-task learning for segmentation and classification of tumors in 3D automated breast ultrasound images, Medical Image Analysis 2021.05

- **Architecture:** Similar to Multi-task learning for segmentation and classification of tumors in 3D automated breast ultrasound images, Medical Image Analysis 2021.05

Experiments & Analysis

1. Implementation Details

- Split ratio is 80%, 20% for the training set, the validation set.
- Epoch: 20
- Optimizer: Adam
- Learning rate: 5e-5
- Weight Decay: 1e-6
- Weights & Biases to track f1, precision, recall, dice, loss

Note: In the self-supervised learning methods, I experimented with amounts of the training data, specifically 100%, 75%, 50%, and 25% of the designated 80% training dataset. Additionally, the first eight layers of the encoder were frozen to evaluate the impact on model performance.

2. Classification

Use all Training set

Observation

- **Dataset Simplicity**: The dataset used is relatively easy.
- **Best Performance**: Swin Transformer has the best performance. However, the performance of most models are nearly.
- **Self-Supervised Learning Efficiency**: The self-supervised approach already develops a robust encoder, thus requiring only the last layers to be fine-tuned. This strategy has 3.08x speedup
- **Minimal Performance Loss**: Self-supervised methods result in only a slight decrease in performance.
- **Zero-Shot Limitations**:
  - The choice of prompts significantly impacts the model's performance.

- The CLIP model, used in a zero-shot learning context, showed some biases as it may not learn specific disease characteristics from the original training data, leading to skewed results.

- **Muti-Task Learning:** Performance slightly decreases, likely due to equal weights (0.5 each) for segmentation and classification losses.

		F1 Score	Precision	Recall	Accuracy	Minute / Epoch
Supervised						
	Swin Transformer	0.9668	0.9699	0.9639	0.9601	8.95
	VIT Base	0.9631	0.965	0.9613	0.9511	8.85
	ConvNeXt	0.9651	0.9663	0.9641	0.9546	7.55
	DenseNet	0.9513	0.9401	0.9644	0.9502	10.65
	EfficientNetV2	0.9654	0.9686	0.9624	0.9561	8.65
	ResNeXt101	0.9597	0.961	0.9268	0.9497	10.3
Self Supervised						
	DINOv2	0.9648	0.9666	0.9602	0.9575	2.45
	BEITv2	0.9611	0.963	0.9593	0.9642	3.95
Zero Shot						
Prompt-1	CLIP	0.4264	0.5843	0.4422	0.6719	
prompt-2	CLIP	0.1689	0.2842	0.2241	0.2255	
Muti-Task						
	VNet	0.9515	0.9589	0.9447	0.9428	7.88

Use 100%, 75%, 50%, 25% training set in Self Supervised method

Observation

- **Performance**: On average, DINOv2 has better performance than BEITv2

- **Effective with Limited Data**: Even with only 50% of the training set, DINOv2 achieves good results, indicating that self-supervised learning is an effective strategy when labeled data is not enough.

		F1 Score	Precision	Recall	Accuracy
100%					
	DINOv2	0.9648	0.9666	0.9602	0.9575
	BEITv2	0.9611	0.963	0.9593	0.9642
75%					
	DINOv2	0.9644	0.9662	0.9626	0.9563
	BEITv2	0.9591	0.9641	0.9548	0.9523
50%					
	DINOv2	0.9614	0.9683	0.9555	0.9544
	BEITv2	0.9559	0.9622	0.9508	0.9494
25%					
	DINOv2	0.9467	0.9435	0.9502	0.9405

		F1 Score	Precision	Recall	Accuracy
	BEITv2	0.9425	0.9484	0.9368	0.9362

3. Segmentation

Observation

- **Best Performance:** DeepLabV3+ has best performance. However, the performance of most models are nearly.
- **Performance**: Muti-task Vnet only resulted in a slight drop in performance.

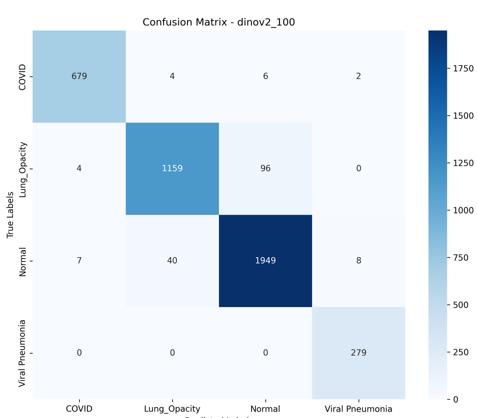
	Val Dice	Val loss
Unet	0.983	0.03628
UNet++	0.9831	0.03711
MAnet	0.9825	0.03786
FPN	0.9825	0.03781
PSPNet	0.9805	0.03394
PAN	0.9822	0.03823
DeepLabV3	0.9818	0.03812
DeepLabV3+	0.9833	0.03451
Muti-Task: VNet	0.9788	

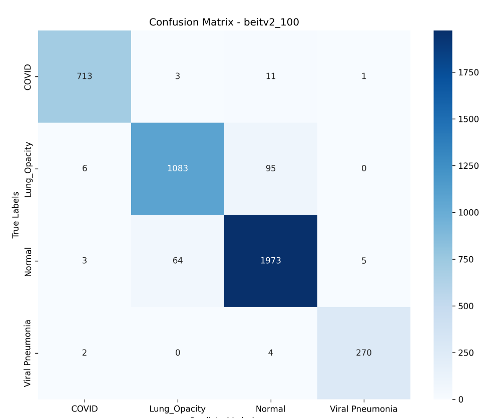
Visualization

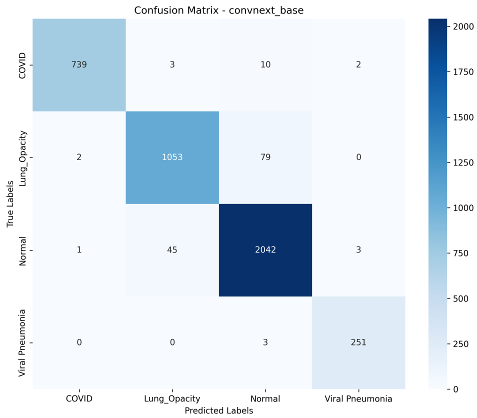
The image below displays the confusion matrices for various models.

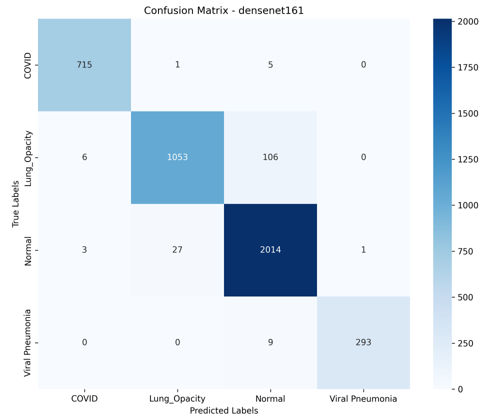
Observation

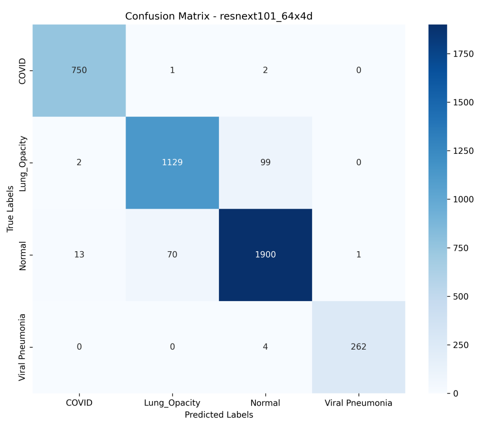
- **Common Confusions**: Most models more frequently confuse Lung Opacity with Normal.
- **Clear Distinctions**: COVID and Viral Pneumonia are generally well-distinguished by the majority of the models.

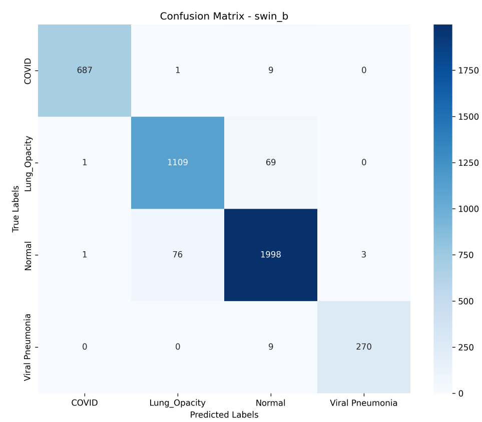


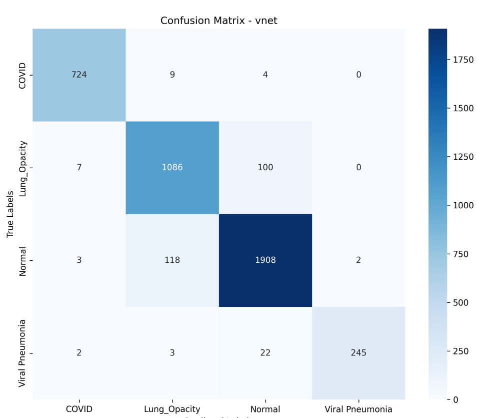


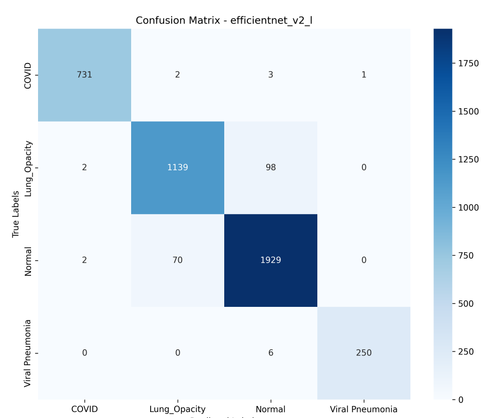

Confusion Matrix - convnext_base
True Labels
Predicted Labels
COVID
Lung_Opacity
Normal
Viral Pneumonia
739 3 10 2
2 1053 79 0
1 45 2042 3
0 0 3 251


Confusion Matrix - densenet161
True Labels
Predicted Labels
COVID
Lung_Opacity
Normal
Viral Pneumonia
715 1 5 0
6 1053 106 0
3 27 2014 1
0 0 9 293


Confusion Matrix - resnext101_64x4d
True Labels
Predicted Labels
COVID
Lung_Opacity
Normal
Viral Pneumonia
750 1 2 0
2 1129 99 0
13 70 1900 1
0 0 4 262


Confusion Matrix - swin_b
True Labels
Predicted Labels
COVID
Lung_Opacity
Normal
Viral Pneumonia
687 1 9 0
1 1109 69 0
1 76 1998 3
0 0 9 270


Confusion Matrix - vnet
True Labels
Predicted Labels
COVID
Lung_Opacity
Normal
Viral Pneumonia
724 9 4 0
7 1086 100 0
3 118 1908 2
2 3 22 245


Confusion Matrix - efficientnet_v2_l
True Labels
Predicted Labels
COVID
Lung_Opacity
Normal
Viral Pneumonia
731 2 3 1
2 1139 98 0
2 70 1929 0
0 0 6 250

Conclusion

In this project, there are the following key points:

- **Many Model Approaches**: Including supervised, self-supervised, zero-shot classification, segmentation, and multi-task learning.
- **Model Performance**: ConvNeXt outperformed others, but overall, many models show close performance levels.
- **Self-Supervised Efficiency**: Fine-tuning only the latter layers significantly sped up training while maintaining strong performance, particularly with DINOv2.
- **Data Efficiency**: Even with reduced data (50% of the training set), DINOv2 provided good results, underscoring the efficacy of self-supervised methods with limited data.
- **Zero-Shot Learning Challenges**: Performance variations with CLIP highlighted the sensitivity to prompts and limitations in learning specific disease features.
- **Multi-Task Learning**: Adjustments to the Unet architecture for simultaneous classification and segmentation.

Reference

Kaggle dataset

torchvision

DINOv2

BEITv2

Unet