Planed for➔ CoATNet + SAM + Grad Cam

Title: CoAtNet with Sharpness-Aware Minimization and Grad-CAM Explainability for Chest X-ray Classification

About CoAtNet Model

=====================

CoAtNet stands for **Convolution and Self-Attention Network.**

Short Note

- we will use either CNN or CoAtNet.
- in this Case CoAtNet is Stroger than CNN.
- Because: CoAtNet has already CNN built inside,
  So, it’s like an upgraded CNN + Transformer hybrid.

Short Note:

CoAtNet combines the strengths of CNNs and Transformers

Practical Example: -

==> If we are classifying **brain tumor MRI images** with only a **few thousand images**:
→ Start with **CNN** (ResNet, EfficientNet).

==>If we are working on a **large chest X-ray dataset** (like CheXpert or MIMIC-CXR with hundreds of thousands of images):
→ **CoAtNet** will likely perform better.

When to choose CoAtNet vs CNN???

=> Small dataset, limited compute

- ➢ Use **CNN** (ResNet, EfficientNet)
- ➢ Reason: simpler, fewer parameters, less likely to overfit.

=> Medium to large dataset

- ➢ Use **CoAtNet**
- ➢ Reason: combines CNN inductive bias with Transformer global power → better accuracy.

=> Very large dataset (millions of samples, like ImageNet-21k, JFT-3B)

- ➢ Use **CoAtNet or pure Transformers**
- ➢ Reason: Transformers (with enough data) are unmatched, and CoAtNet bridges the gap.

Types of CoAtNet :

There are Multiple use of CoAtNet like CoAtNet-0, CoAtNet-1, CoAtNet-2, CoAtNet-3, CoAtNet-4, CoAtNet-5

[So in our Case ---> we used **coatnet-0**]
Because, it By default, the most practical choice

Short Note

If dataset is larger and GPU is strong, we can go for coatnet_1 or coatnet_2

About SAM Technique

==========================

- SAM stands for Sharpness-Aware Minimization.
- It's an optimization technique (not a new architecture or Model) introduced by Google Research in 2021.
- Goal: Make the model prefer flat minima in the loss

Short Note:

- ➢ It is an **optimization technique,** not a new network or Model.
- ➢ We apply SAM in place of a **standard optimizer** (like SGD/Adam).

Where do we use the optimizer in CoAtNet?

- ➢ The **model** (CoAtNet) = just the layers and **parameters** (convs, attention, etc.).
- ➢ The **optimizer** = external **component that updates** those **parameters** during **training.**
- ➢ So, the optimizer is applied after the **forward + backward** pass of CoAtNet.

What exists by default (without SAM)?

With SAM, we wrap the base optimizer (SGD/AdamW).

- ❖ Normally, if you train CoAtNet:
- ❖ We can choose Adam / AdamW / SGD as the optimizer

python Copy code

```python
optimizer = torch.optim.AdamW(coatnet.parameters(), lr=1e-4, weight_decay=1e-4)
```

Explain:

This means the weights in all stages of CoAtNet (conv layers + attention layers) are updated using AdamW.

What changes when you use SAM?

- ❖ With SAM, you wrap the base optimizer (SGD/AdamW).
- ❖ Instead of plain AdamW, you use:

python Copy code

```python
from sam import SAM

base_optimizer = torch.optim.AdamW
optimizer = SAM(coatnet.parameters(), base_optimizer, lr=1e-4, weight_decay=1e-4)
```

Explain:
Now, during training, SAM performs two steps:
(1) Perturb weights to find “sharp” regions.
(2) Update weights toward flat minima.

🔹 **Key Takeaway**

- Optimizer section in CoAtNet training = where we normally use Adam/SGD.
- Without SAM → just standard optimizers (SGD, Adam, AdamW, RMSProp, etc.).
- **With SAM** → those optimizers are **wrapped inside SAM,** making training more **robust.**

👉 So SAM does not change CoAtNet layers; **it only changes how the parameters are updated.**

Short Note

- ❖ SAM improves generalization and robustness by training models to avoid sharp minima and converge toward flat minima.
- ❖ Works well across CNNs, Transformers, and hybrids (like CoAtNet).

Why SAM is Stronger than Normal Optimizers:

- ➢ Standard training → can fall into sharp minima (low training loss but poor generalization).
- ➢ SAM → explicitly finds worst-case loss in a neighborhood around weights and minimizes it.
- ➢ This forces the model to learn flatter solutions, which generalize better.

Practical Example:

==> If training on a small dataset:
→ Normal optimizers (Adam/SGD) may overfit → SAM helps by improving generalization.

==> If training on a large dataset:
→ SAM makes the model more robust, often achieving state-of-the-art accuracy compared to baseline optimizers.

When to use SAM??

=> If baseline CNN/Transformer is overfitting

- ❖ Use SAM
- ❖ Reason: It smooths the loss landscape and prevents sharp minima.

=> If working with noisy medical data (like chest X-rays)

- ❖ Use SAM

- ❖ Reason: Improves robustness and reduces overfitting to dataset-specific noise.

=> If model already performs well but you want extra generalization boost

- ❖ Use SAM
- ❖ Reason: Often improves accuracy by 1–2% without architecture changes.

In Our case (Chest X-ray dataset):

- o We already use CoAtNet-0 as backbone (great choice for medium dataset).
- o Adding SAM as optimizer → will improve generalization and reduce risk of overfitting.
- o Then, applying Grad-CAM → gives explainability (why the model predicted TB/COVID/etc.).

pipeline will be:

=> **CoAtNet-0 (feature extractor) + SAM (optimizer) + Grad-CAM (explainability)** <=

Short Note

So, when we say "Hybrid CoAtNet + SAM", that's a little misleading, since CoAtNet is the hybrid, and SAM is an optimizer that makes training more generalizable.

- **A better way to phrase it:**
- We train CoAtNet using the SAM optimizer and explain it with Grad-CAM.

- **SO, Best description of our research is:**
- 👉 "We propose a CoAtNet model trained with SAM optimizer and interpreted using Grad-CAM for chest X-ray classification."

==== > 🔹 Final Setup for our Chest X-ray project <======
==

- ❖ **** **Model backbone**: CoAtNet-0 (or CoAtNet-1/2 if GPU allows) **** [Hybrid architecture(CoAtNet) = strong feature learning.]

- ❖ ******Training optimizer**: SAM (instead of vanilla SGD/Adam) **** [SAM optimizer = robustness + generalization.]

- ❖ ******Explainability:** Grad-CAM (to visualize regions influencing predictions) **** [Grad-CAM = interpretability (critical in medical imaging).]

How we proceeded step by step In Model
==

Step1: Model Selection

- CoAtNet-0 (via timm), pretrained on ImageNet.
  Adapted for Our number of classes (Pneumonia, Normal).
- We chose CoAtNet-0 because:
  - It's a hybrid CNN + Transformer (stronger than plain CNN for medical images).
  - Smaller size → avoids overfitting since your dataset is not as huge as CheXpert/MIMIC-CXR.

Step 2: Data Preparation

- Grayscale → 2 channels.
- Resized to 224×224, normalization with ImageNet mean/std.
- Augmentations: random horizontal flip for training. train_loader and test_loader.

Step 3: Training Setup

- Loss: CrossEntropyLoss.
- Optimizer: SAM implemented (custom wrapper).
- Base optimizer: SGD (momentum=0.9, weight decay
- Scheduler: StepLR.

Short Note: We originally trained with **Adam in SAM**

Step 4: Training loop

- Full two-step SAM training loop implemented (first_step, second_step).
- Model checkpoints saved when val acc improves (best_coatnet_sam.pth).

Step 3: Evaluation

- Reload best weights
- Test accuracy printed.
- Classification report + confusion matrix.
-

Step 3: Explainability (Grad-CAM)

- Visualized heatmaps overlayed on original chest X-rays.
- Allows you to see which lung regions influenced predictions.