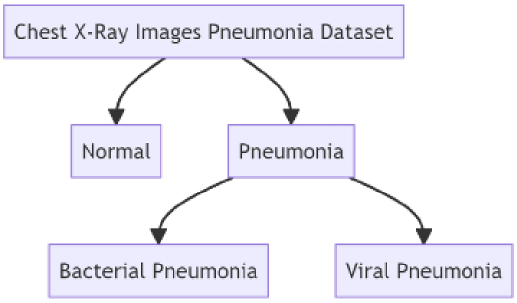
Dataset Overview and Pre-process

Slide 11 Dataset Overview

Data & Experiment



Process

Type	Train Set	Test Set
Normal	1341	234
Bacterial	2530	242
Viral	1345	148
Total	5216	624

Datase

The dataset used in this study is the “Chest X-Ray Images (Pneumonia)” dataset, provided by Paul Timothy Mooney and available on Kaggle.

This dataset is specifically designed for the task of pneumonia detection and classification, offering a substantial resource for training and evaluating medical image analysis models.

> 在医学图像分析领域，胸部 X 光片的自动诊断具有至关重要的意义。本研究使用的数据集是“Chest X-Ray Images (Pneumonia)”数据集，由 Paul Timothy Mooney 提供，并可在 Kaggle 上获得。该数据集专门用于肺炎检测和分类任务，为医学图像分析模型的训练和评估提供了重要资源。

Slide 12 Data Pre-processing

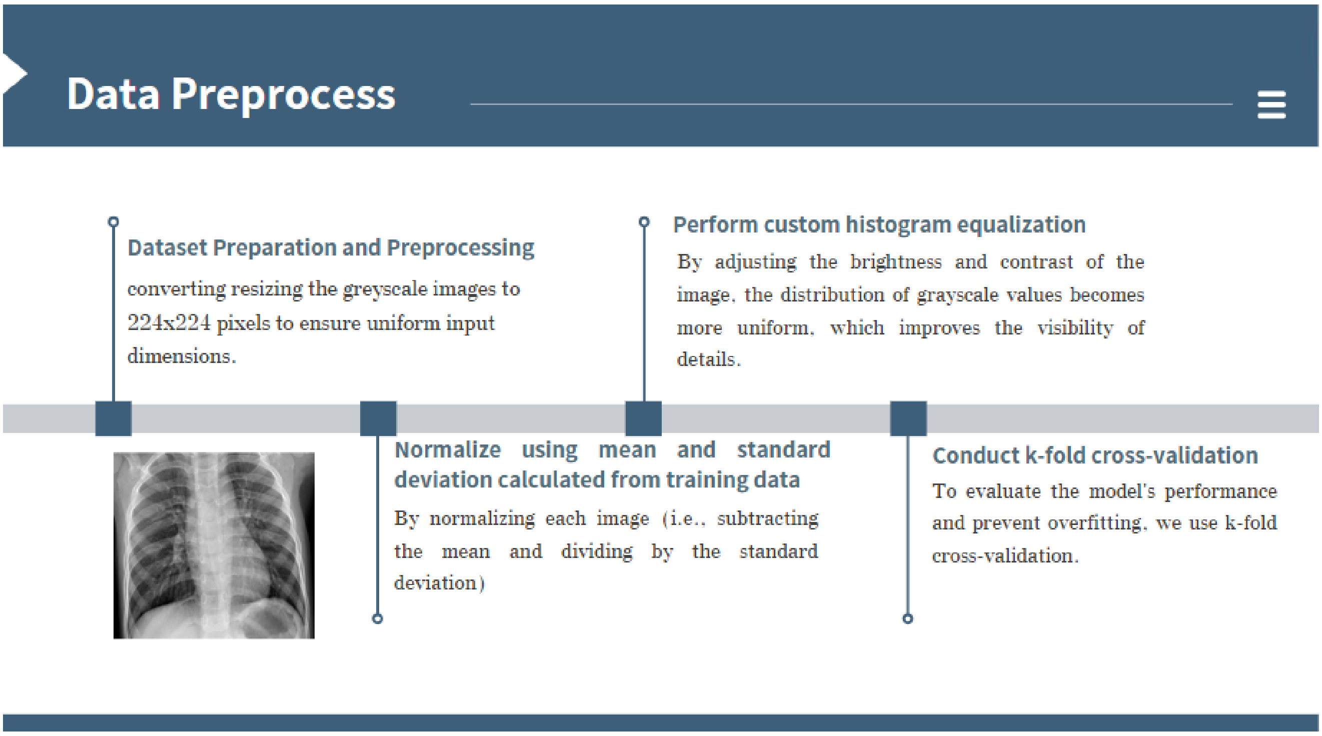


Image preprocessing is a crucial step to ensure the effectiveness of model training and to improve prediction accuracy.

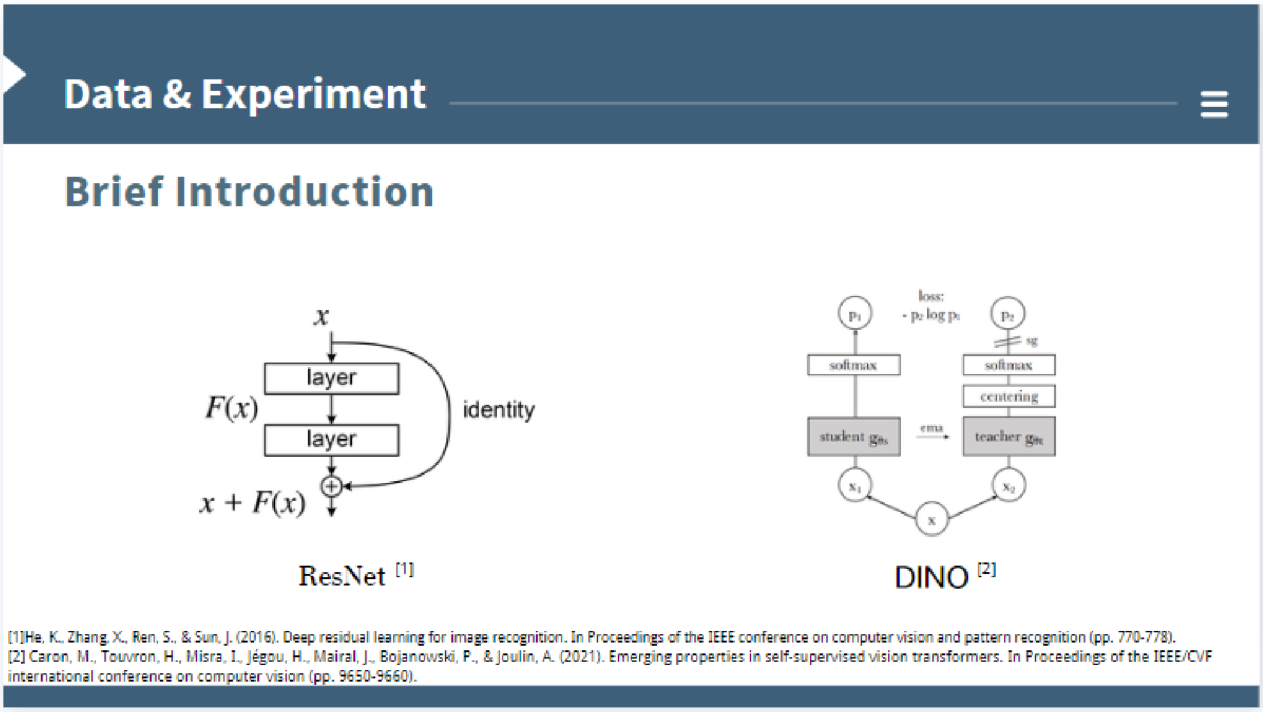
Below is a detailed description of the image preprocessing steps:

1. **Resize to 224 x 224 pixels**
2. **Normalize using mean and standard deviation calculated from training data**
3. **Perform custom histogram equalization**
4. **Conduct k-fold cross-validation**

Model

Let's move to model building.

Slide 13 Brief Intro to Model



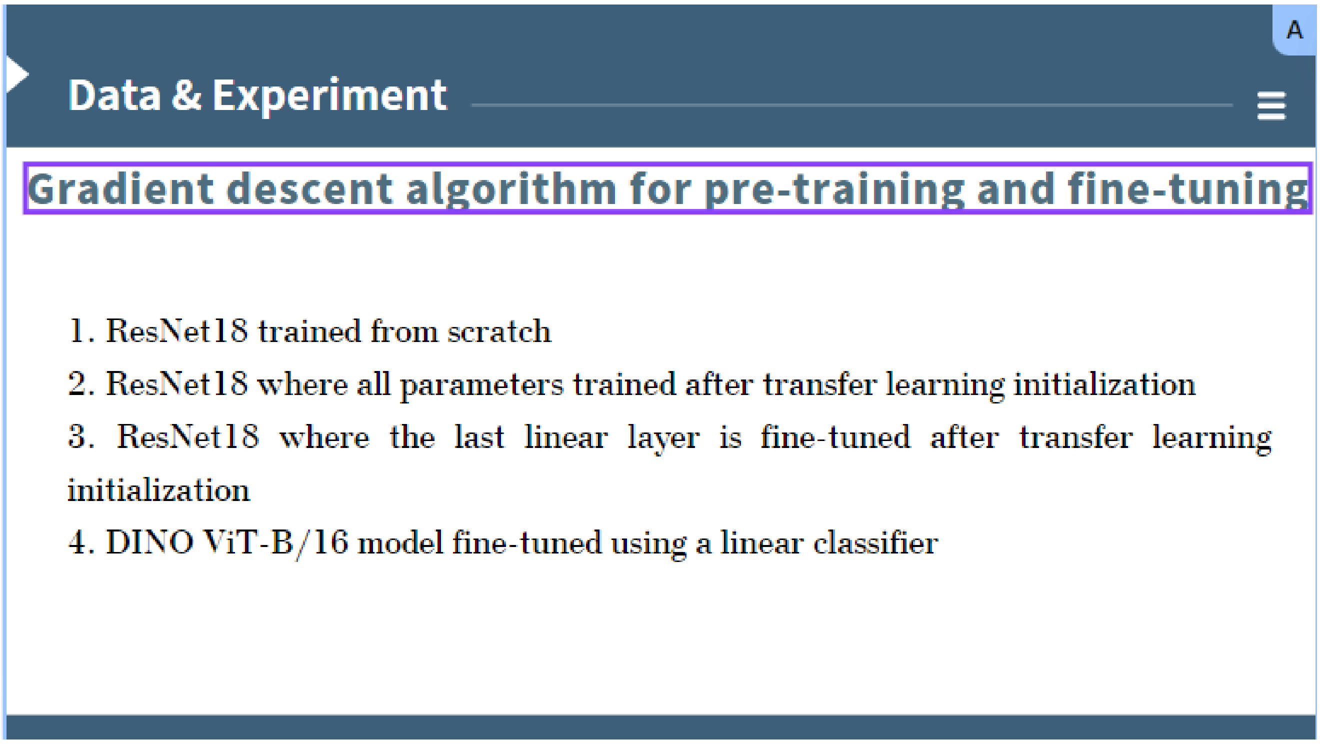
ResNet

ResNet is a deep convolutional neural network that uses residual connections to alleviate the vanishing gradient problem, enabling the effective training of very deep networks. It excels in image classification and computer vision tasks, with various versions such as ResNet-50 and ResNet-101.

DINO ViT-B/16

DINO ViT-B/16 is a self-supervised learning method that uses Vision Transformers (ViT) to learn image features without requiring labeled data. The ViT-B/16 variant employs 16x16 pixel patches to capture global dependencies in images.

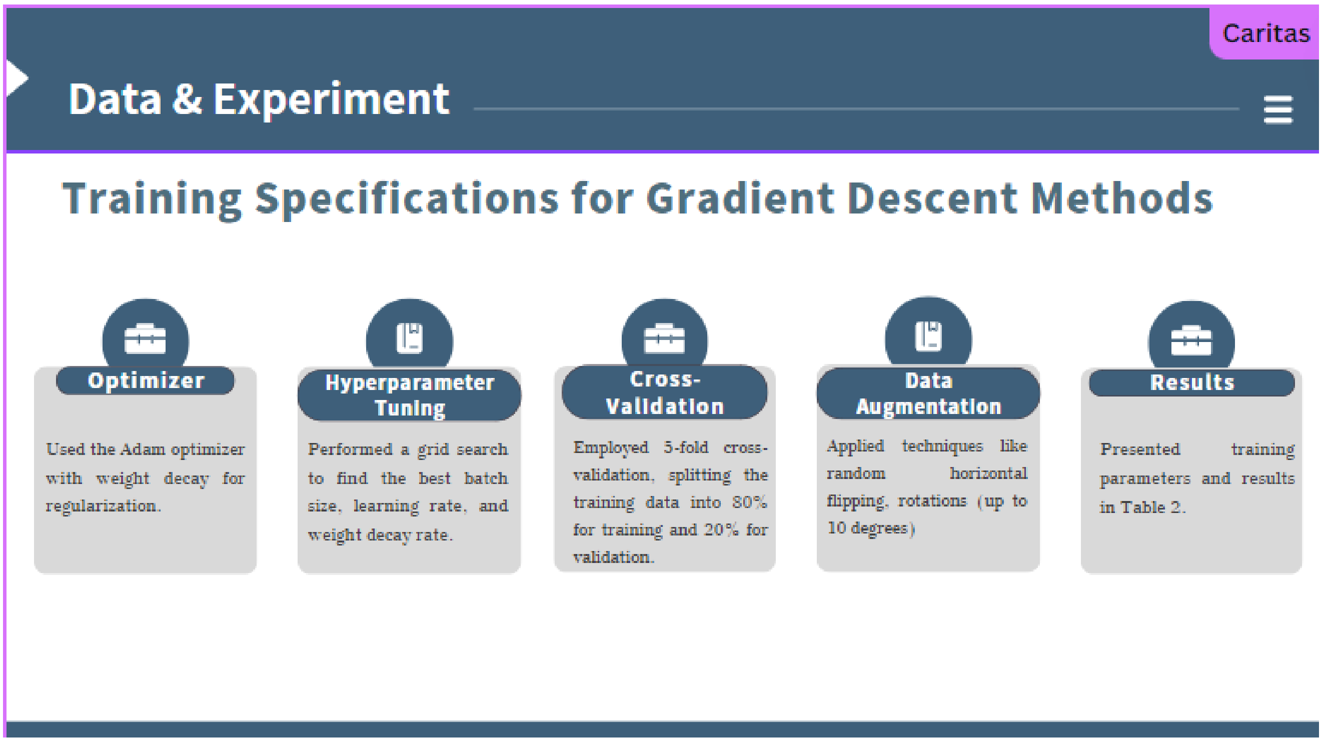
Slide 14 Gradient descent algorithm for pre-training and fine-tuning



Using gradient descent algorithm, we have trained following model in our dataset:

1. ResNet18 trained from scratch
2. ResNet18 where all parameters trained after transfer learning initialization
3. Resnet18 where the last linear layer is fine-tuned after transfer learning initialization
4. DINO ViT-B/16 model fine-tuned using a linear classifier.

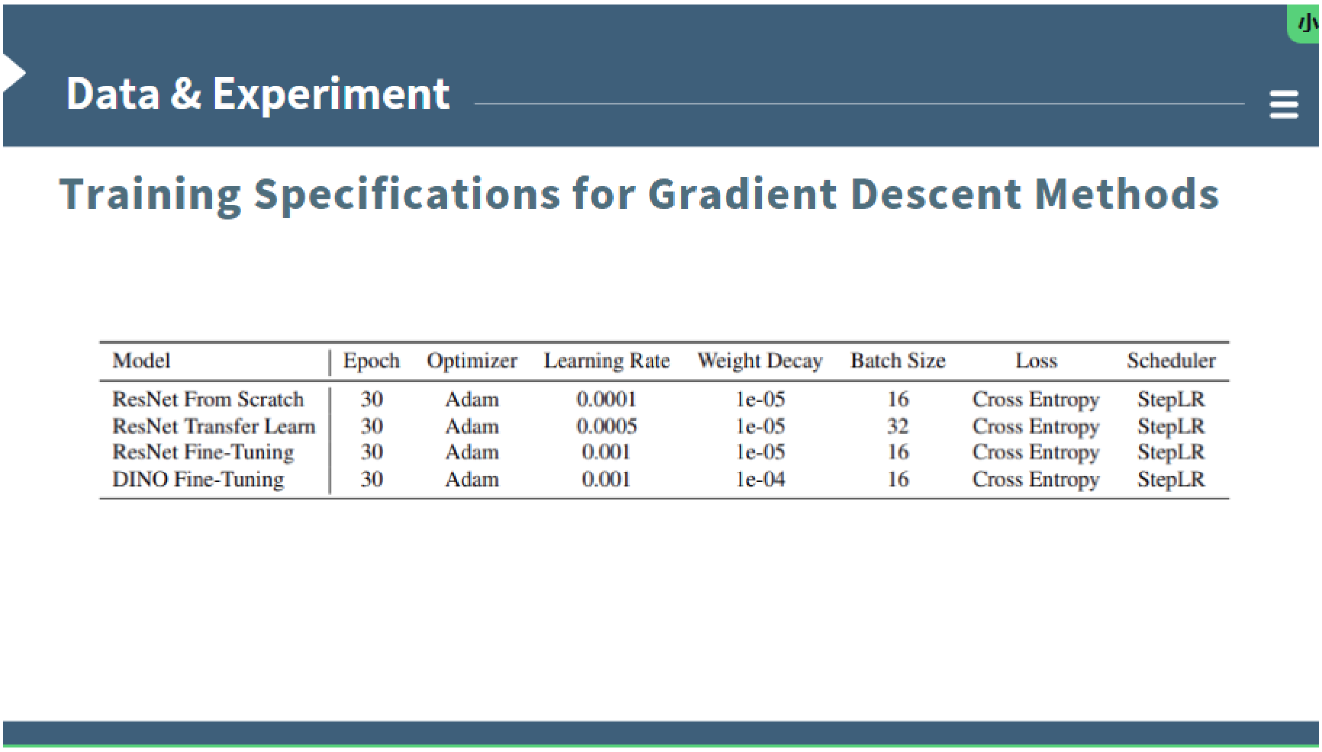
Slide 15 Training Specifications for Gradient Descent Methods



For training the models, we utilized the Adam optimizer with weight decay for regularization.
To determine the optimal hyperparameters, we conducted a grid search to find the best values for batch size, learning rate, and weight decay rate.
Each combination was evaluated using k-fold cross-validation with (k = 5).

Given the imbalanced class distribution in the data, we applied data augmentation techniques during each train-validation split.
These techniques included random horizontal flipping, random rotations (up to 10 degrees), and Gaussian blur.
The training parameters obtained from the grid search and 5-fold cross-validation are summarized in Table 2.

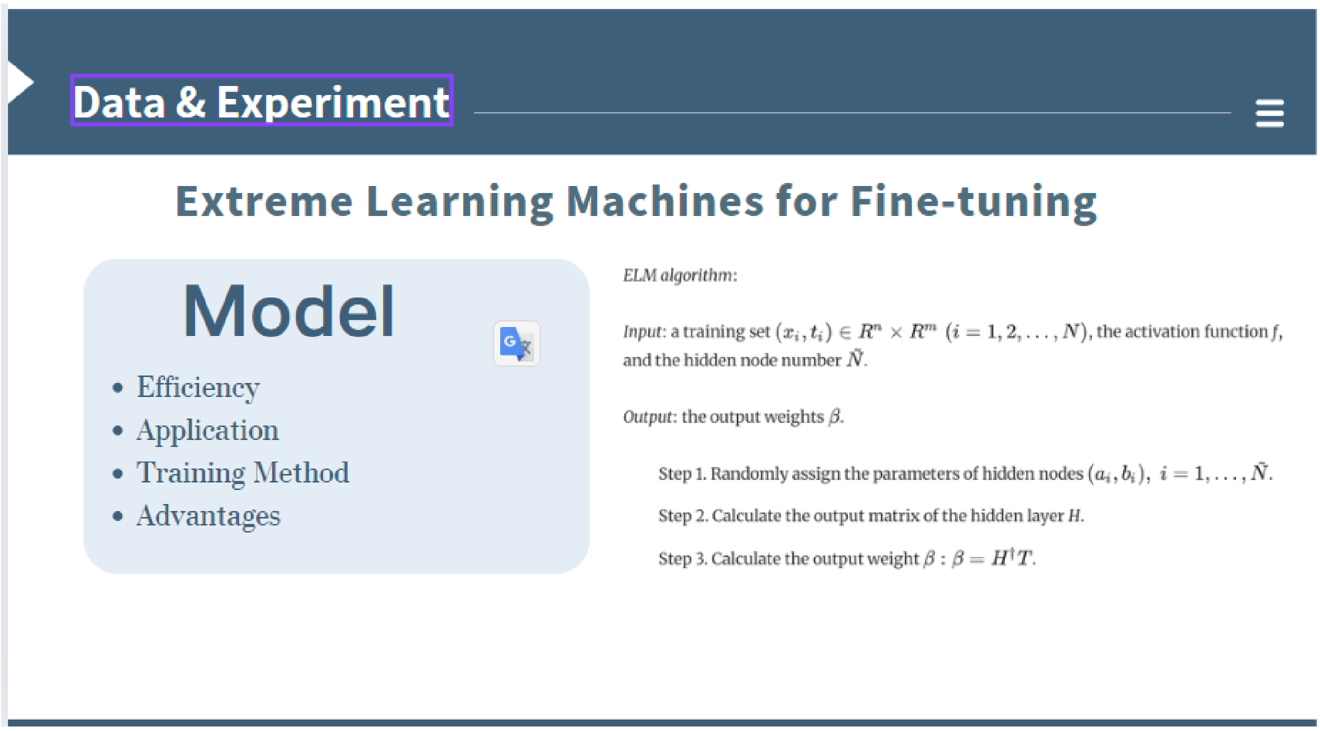
Slide 16 Hyperparameter



Model	Epoch	Optimizer	Learning Rate	Weight Decay	Batch Size	Loss	Scheduler
ResNet From Scratch	30	Adam	0.0001	1e-05	16	Cross Entropy	StepLR
ResNet Transfer Learn	30	Adam	0.0005	1e-05	32	Cross Entropy	StepLR
ResNet Fine-Tuning	30	Adam	0.001	1e-05	16	Cross Entropy	StepLR
DINO Fine-Tuning	30	Adam	0.001	1e-04	16	Cross Entropy	StepLR

Let me show you the best hyperparameter of the models we found.

Slide 17 ELM



Now, let me introduce ELM.

Extreme Learning Machines (ELMs) [7] provide a rapid and efficient approach for training linear classifiers, especially when applied to features obtained from self-supervised models.

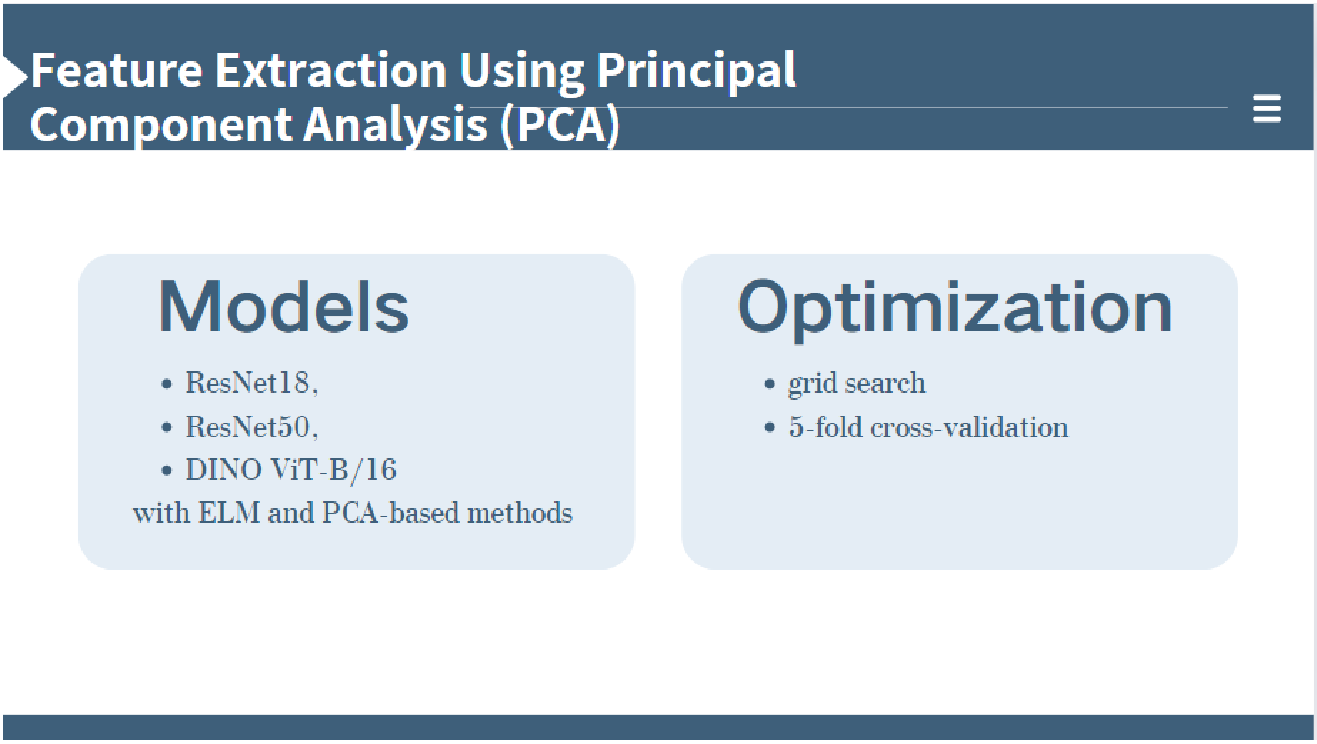
Unlike conventional neural networks that depend on iterative backpropagation, ELMs use a simple closed-form solution to compute the output weights.

This method greatly decreases computational complexity and training time.

How do we employ ELM in our project?

Slide 18 PCA

Before PCA: Extract several features for training a linear classifier with ELM.



Models

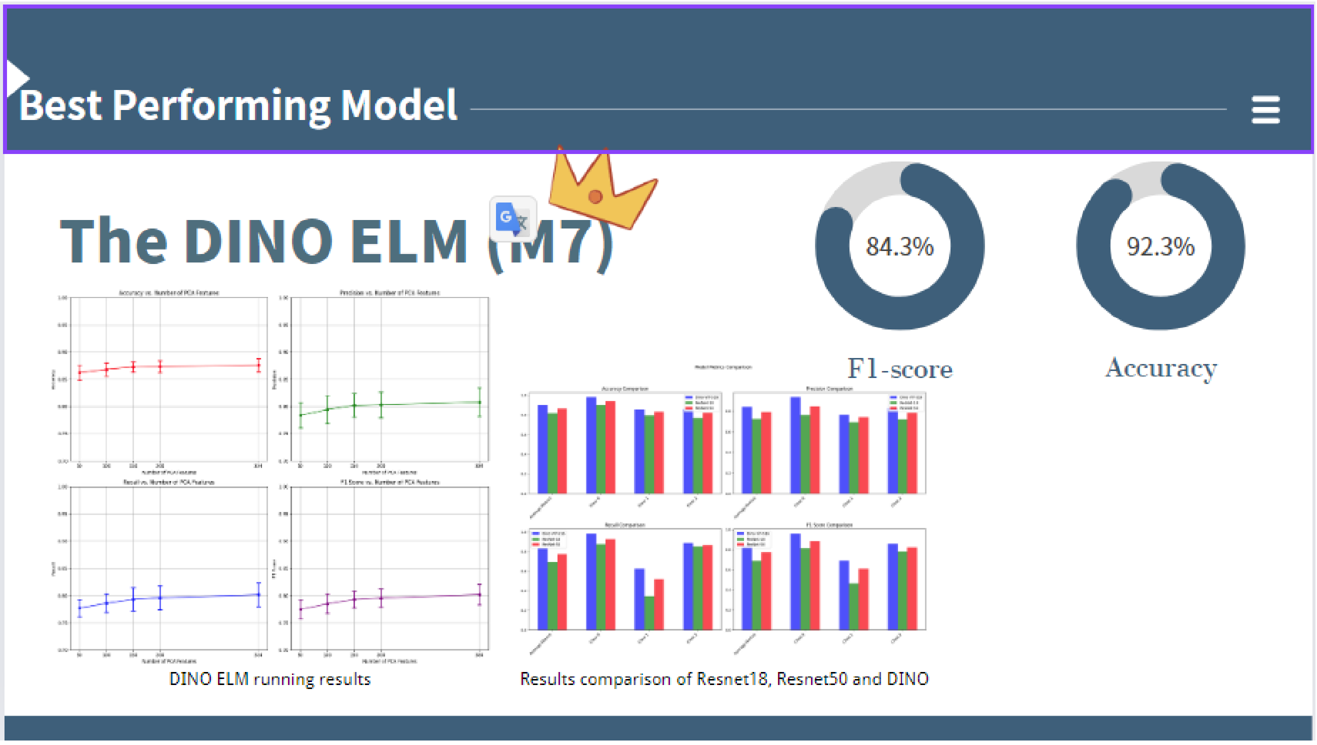
We employed our ELM and PCA-based fine-tuning approach to train linear classifiers using the following models.

Optimization

To optimize, we conducted a grid search combined with 5-fold cross-validation.

Result (Slide 19)

Slide 20 Best Performing Model

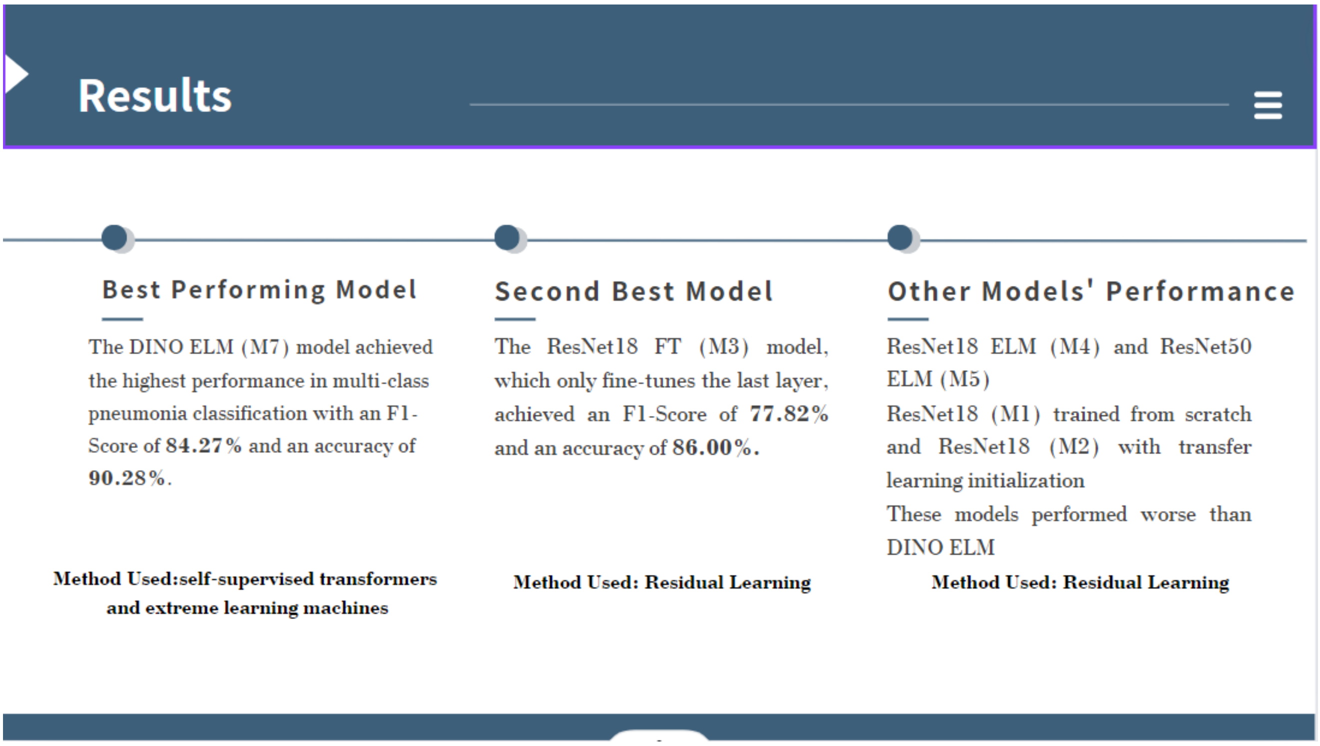


The DINO ELM (M7) model achieved the highest performance in multi-class pneumonia classification with an F1-Score of 84.27% and an accuracy of 90.28%.

This model combines self-supervised transformers and extreme learning machines, demonstrating strong generalization capabilities.

Slide 21 Results

And this is the rest of model we test.



The second best is Fine-tuned ResNet18 by Microsoft.

Let's move to CONCLUSION & FUTURE ANALYSIS