HUTECH
Đại học Công nghệ Tp.HCM

Chung kết Cuộc thi

OLYMPIC TRÍ TUỆ NHÂN TẠO - OAI HCMC 2025

(Olympiad in Artificial Intellgence at Ho Chi Minh City 2025)

TP. Hồ Chí Minh, ngày 25 tháng 04 năm 2025

MUSHROOM CLASSIFICATION

Presenter: EEIoT_newbie

VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

EEIoT_newbie

VĂN THỊNH

DUY TÂN

ANH QUÂN

MINH HUY

HCMUT EE
Machine Learning & IoT Lab

Outline

- 1. Introduction
- 2. Problem Statement
- 3. Method
- 4. Experimental Results

1. Introduction

1. Introduction

Topic: Classification of dried mushroom images.

Objective: Build a machine learning model to classify images into three categories:

- **Shiitake Mushroom** (label 0)
- **Wood Ear Mushroom** (label 1)
- **Snow Fungus** (label 2)

Dataset:

- Color images, **32x32 pixels** resolution.
- **Training set**: 1050 images.
- **Testing set**: 450 images.

2. Problem Statement

2. Problem Statement

Input:

- **Training**:
  - 500 RGB color images, size **32x32 pixels**, equally distributed among: Snow Fungus (167), Wood Ear Mushroom (166), Shiitake Mushroom (167).
  - All images have uniform dimensions, 1:1 aspect ratio, file size between **0.80 – 1.04 KB**.
- **Inference**:
  - 450 unlabeled test images with the same resolution and RGB format.
  - Statistical differences in **brightness** and **contrast** across classes:
    - Shiitake: brightest (114.09), lowest contrast (41.13)
    - Snow Fungus: moderate brightness (102.14), medium contrast (52.84)
    - Wood Ear: darkest (101.63), highest contrast (59.09)

Output: Predict the correct label (class) for each test image:

- **0**: Shiitake Mushroom
- **1**: Wood Ear Mushroom
- **2**: Snow Fungus

3. Method

3. Method

❖ Calculating Mean and Standard Deviation for Data Normalization

Why Normalize?

Helps deep learning models **train faster and more stably**.
Ensures all input features (pixels) are on a similar scale.
Typically transforms image pixel values so each channel (RGB) has **mean = 0** and **standard deviation = 1**.
Prevents issues like exploding or vanishing gradients.

3. Method

Calculating Mean and Standard Deviation for Data Normalization

Theoretical Formulas:

- Mean ($\boldsymbol{\mu}$) per channel:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

- Standard Deviation (σ) per channel:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (x_i - \mu)^2}$$

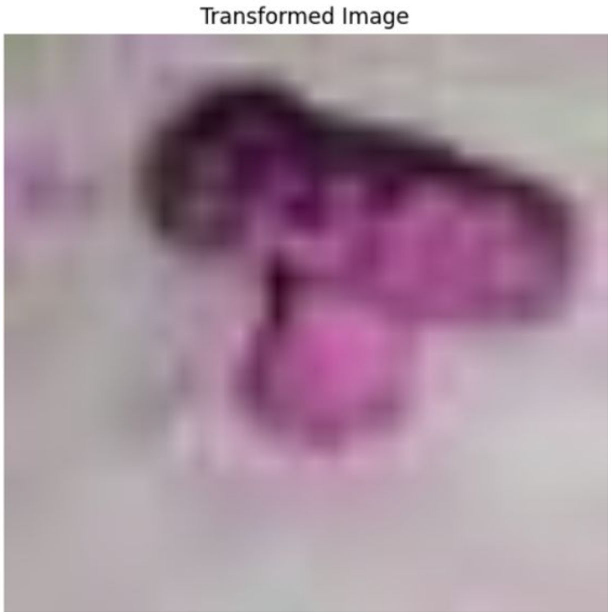
Where x_i is a pixel value, and N is the total number of pixels across all images per channel.

3. Method

- ❖ **RandomHorizonntalFlip**
- ❖ **RandomVerticalFlip**
- ❖ **ResizeCrop**

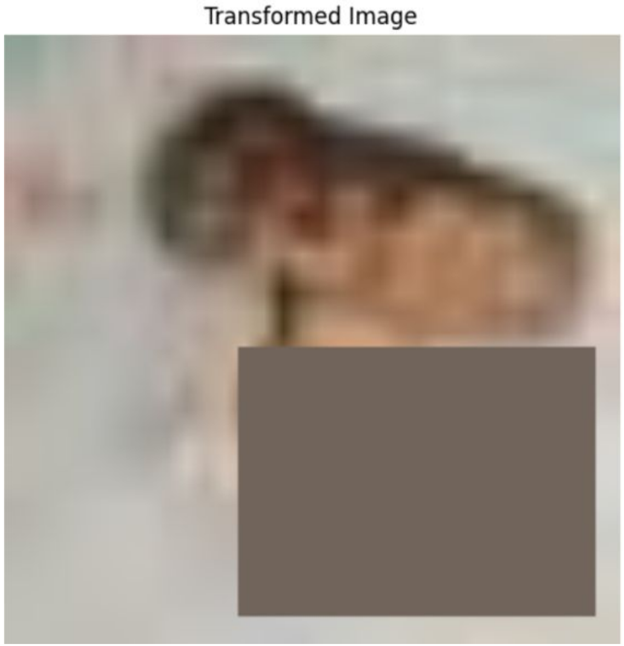
3. Method

❖ ColorJitter



3. Method

❖ RandomErasing (Cutout)

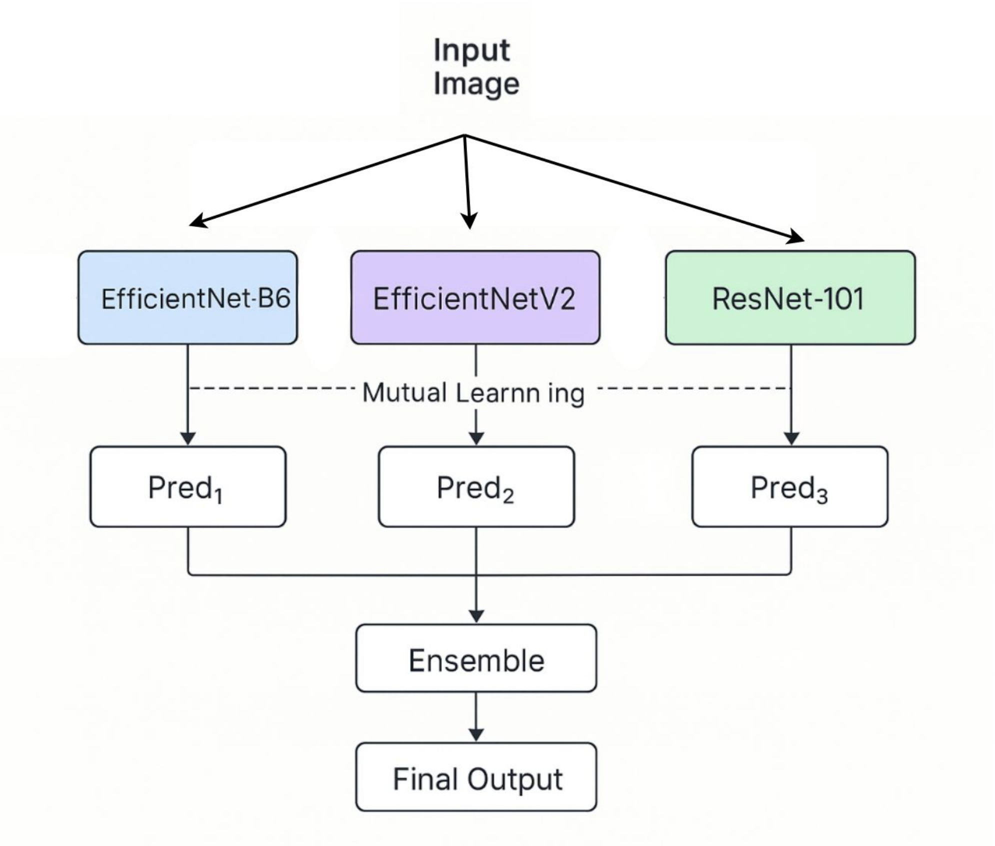


3. Method

❖ Elastic Transform

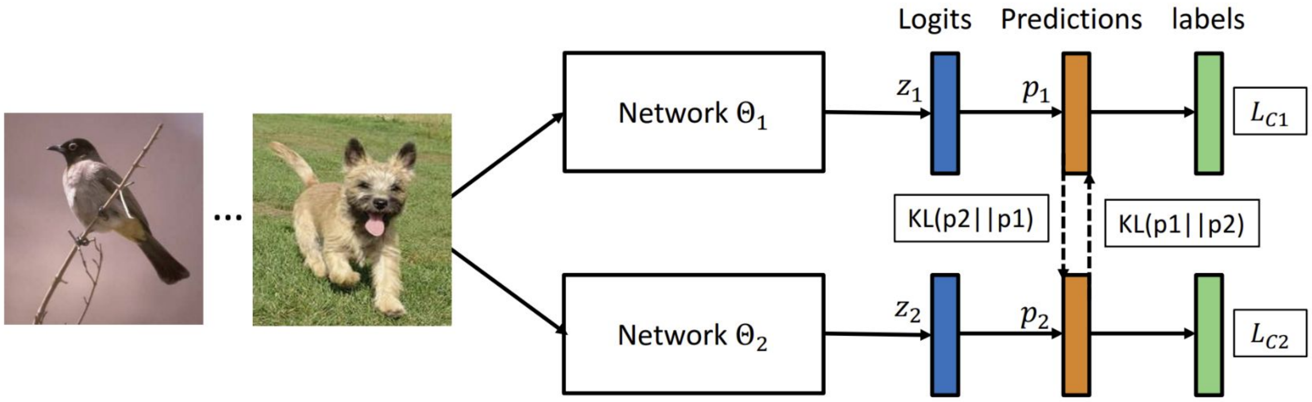
3. Method

❖ Overview



3. Method

❖ Deep Mutual Learning



3. Method

❖ Deep Mutual Learning

4. The total loss for model Θ_i with K networks:

$$L_{\Theta_i} = L_{C_i} + \frac{1}{K-1} \sum_{k=1, k \neq i}^{K} D_{KL}(p_k \| p_i)$$

$$L_{C_i} = -\sum_{i=1}^{N} \sum_{m=1}^{M} I(y_i, m) \log(p_i^m(x_i))$$

Cross-Entropy

$$D_{KL}(p_k \| p_i) = \sum_{i=1}^{N} \sum_{m=1}^{M} p_k^m(x_i) \log \frac{p_k^m(x_i)}{p_i^m(x_i)}$$

Kullback-Leibler (KL) divergence

3. Method

❖ Ensemble Learning with Unweighted Averaging

$$p = \frac{1}{K}\sum_{k=1}^{K} p_k$$

K : the number of models

p_k : the probabilities are calculated from model **k**

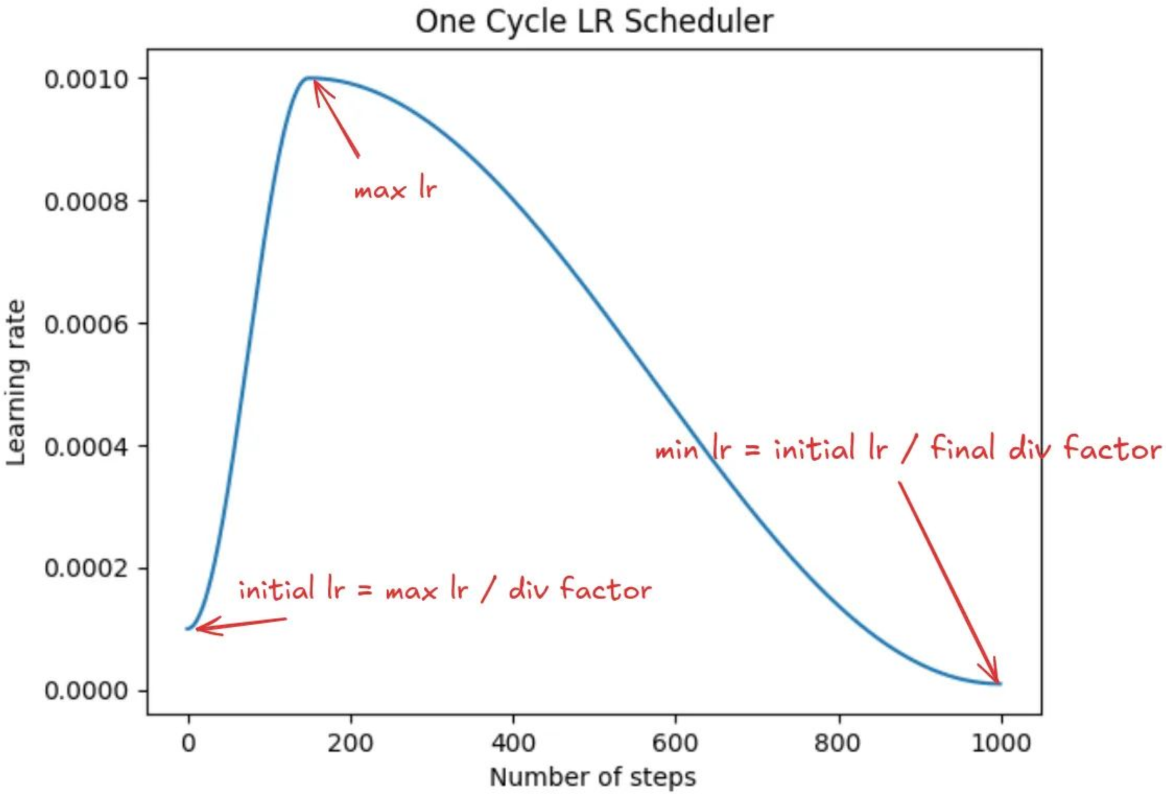
4. Experimental Results

4. Experimental Results

- **Image size :** 224 x 224
- **Batch size :** 32
- **LossFunction :** CrossEntropyLoss(label_smoothing=0.1)
- **Optimizer :** AdamW(lr=0.001, weight_decay=0.01)
- **EarlyStopping :** finish training when valid_acc does not improve (3 patience)
- **clip_grad_norm :** normalize gradient

4. Experimental Result

- **OneCycleLR**



4. Experimental Result

❖ GradScaler - Mixed Precision Training

Use float16 instead float32 gradient

- Scale loss : multiplying gradient to ***scale factor***-> avoid vanishing gradient
- Unscale loss
- Adjust scale factor flexibly

=> Faster training

=> Optimize GPU utilization

4. Experimental Result

❖ Public Test

Method	Num_epochs	Optimizer	DML	Ensemble	Test Accuracy
Resnnet-34	30	SGD	✗	✗	0.76
ViT-L16	20	SGD	✗	✗	0.83
GoogleNet	20	Adam	✗	✗	0.85
MobileNet	15	Adam	✗	✗	0.85
EfficientNetB6-CBAM	20	SGD	✗	✗	0.91
EfficientNetB6	17	SGD	✗	✗	0.96
MobileNet + EfficientNetB6 + VGG16	50	Adam	✗	✓	0.95
EfficientNetB6* + EfficientNetV2 + Resnet101	15	AdamW	✓	✗	0.95
EfficientNetB6 + EfficientNetV2 + Resnet101	15	AdamW	✓	✓	**0.98**

4. Experimental Result

❖ Private Test

Method	Num_epochs	Optimizer	DML	Ensemble	Test Accuracy
EfficientNetB6 + EfficientNetV2 + Resnet101	15	AdamW	✓	✓	**0.96**

			Accuracy	F1-Score
4	OAI069	EEIoT_newbie	**96.17%**	**96.10%**

4. Experimental Result

❖ Private Test (Final)

Method	Num_epochs	Optimizer	DML	Ensemble	Test Accuracy
EfficientNetB6 + EfficientNetV2 + Resnet101	15	AdamW	✓	✓	**0.96**