Artificial Intelligence – Week 4 (1st Mini Project)

Instructor: Teerapong Panboonyuen

Course Repository:
https://github.com/kaopanboonyuen/SC310005_ArtificialIntelligence_2025s1

🎯 Objective

This week, you will apply supervised machine learning techniques to an image classification problem. The task involves:

- Finding a dataset for image classification (can be domain-specific or general, like CIFAR-10, MNIST, etc.).
- Designing your custom architecture to classify images.
- Implementing and benchmarking your custom model against three state-of-the-art (SOTA) models:
  - **ResNet**
  - **DenseNet**
  - **Vision Transformer (ViT)**

Your goal is to compare your custom architecture's performance with the three SOTA models, using metrics like accuracy, loss, and training time.

🧪 Let's Do the Lab!

🎯 **Project Goal**: Classify faces of 10 Thai Prime Ministers using deep learning!

Model Zoo:

https://docs.pytorch.org/serve/model_zoo.html

📚 Dataset Description

The dataset you choose should contain images with corresponding labels for classification tasks. You can find datasets in popular repositories like:

- Kaggle
- UCI Machine Learning Repository
- Google Dataset Search

For example, you could use datasets such as:

- **CIFAR-10/CIFAR-100**: Image classification with 10/100 categories.
- **Fashion MNIST**: Image dataset of fashion products.
- **Custom Dataset**: Medical images, satellite images, etc.

Ensure your dataset is appropriate for classification and contains sufficient data for training and validation.

📝 Assignment Instructions

1. Dataset Selection:

- **Choose a dataset** related to image classification that fits your interests. Ensure the dataset contains images and labels.

2. Data Preprocessing:

- Preprocess the images as needed:
    - Resize or crop the images.
    - Normalize pixel values (e.g., scaling to [0, 1] range).
    - Optionally, apply data augmentation (rotation, flipping, etc.).

3. Custom Model:

- **Design and implement your custom image classification architecture** using convolutional layers, fully connected layers, etc.
    - Experiment with different layer structures, activation functions, and regularization techniques (e.g., dropout).

4. Baseline with SOTA Models:

- Implement and train the following three state-of-the-art models:
    - **ResNet** (Residual Networks)
    - **DenseNet** (Densely Connected Convolutional Networks)
    - **Vision Transformer (ViT)** or any other transformer-based model

5. Model Training:

- Train all models on the dataset.
- Fine-tune hyperparameters (learning rate, batch size, optimizer, etc.) to improve performance.

6. Evaluation:

- Evaluate all models on the test set.
- Use metrics like **accuracy**, **precision**, **recall**, **F1 score**, and **confusion matrices** for evaluation.

7. Performance Comparison:

- Compare the performance of your custom model with the three SOTA models (ResNet, DenseNet, ViT) using:
  - Accuracy and Loss Curves.
  - Visualizations such as confusion matrices and ROC curves.

8. Bonus (Optional):

- Use data augmentation techniques to improve model performance.
- Experiment with pre-trained models and transfer learning.

🔥 Example Approaches:

Here are some strategies you can try:

- **ResNet**: Known for its skip connections that help mitigate vanishing gradients.
- **DenseNet**: Uses dense connections between layers, promoting feature reuse and improving accuracy.
- **Vision Transformer (ViT)**: A transformer-based model that treats images as sequences of patches, widely regarded as a powerful alternative to

CNNs for certain image classification tasks.

💡 Deliverables:

1. **Jupyter Notebook or Google Colab File**:
    - Code implementing data preprocessing, model architectures, and training.
    - Clear explanations for each step of the process, including your rationale for choosing certain models or architectures.
    - Visualizations for model performance comparison (accuracy/loss curves, confusion matrices, ROC curves).
    - A **summary table** comparing the performance of all models.
2. **Documentation**:
    - Write a summary of your approach, how you trained your models, and what insights you gained from the performance comparison.

🎯 Evaluation Criteria:

Your assignment will be evaluated based on the following:

- **Correctness and completeness** of your ML pipeline (data preprocessing, model training, evaluation).
- **Effectiveness** in improving the baseline performance (accuracy, loss, etc.).

- **Data preprocessing**: Correct handling of image data, resizing, normalization, and augmentation (if used).
- **Model performance comparison**: Accuracy, loss, confusion matrix, and other relevant metrics.
- **Quality of explanations and code readability**.
- **Creativity** and **advanced techniques** (e.g., transformer models, transfer learning).
- **Clear and insightful visualizations** (accurately interpreting confusion matrices, ROC curves, etc.).

🚀 Getting Started:

1. **Dataset Selection**: Choose a dataset for image classification from any of the available sources mentioned above. Ensure it fits the classification task.
2. **Model Implementation**:
   - Implement your custom model using TensorFlow/Keras or PyTorch.
   - Implement the baseline models: ResNet, DenseNet, and Vision Transformer (ViT).

📅 Submission Deadline:

The final submission date will be announced and agreed upon during class. Please ensure you complete all tasks and submit the Jupyter notebook or Google Colab file by the due date.

Good luck, and I look forward to seeing your innovative architectures and performance comparisons!

Sample 1: Flower Species Classification at Khon Kaen University

---

Objective:

Create a custom deep learning model to classify different species of **flowers** found around **Khon Kaen University (KKU)**.

Steps:

1. **Dataset**:
   - Collect images of flowers in different parts of KKU campus.
   - Label flowers by species (e.g., Bougainvillea, Orchid, Hibiscus, etc.).
2. **Custom Model**:
   - Design a CNN with multiple convolutional layers for feature extraction.
   - Use pooling and dense layers for classification.
3. **Baseline Models**:
   - Compare with **ResNet**, **DenseNet**, and **Vision Transformer (ViT)**.
4. **Goal**:
   - Train and evaluate the model's accuracy and performance on unseen images of flowers.

Sample 2: Fruits Classification at a Local Market

Objective:

Build a custom deep learning model to classify different types of fruits sold at a **local market** near your area.

Steps:

1. **Dataset**:
    - Capture images of fruits like **Apple**, **Banana**, **Orange**, and **Papaya** in different lighting conditions and angles.
2. **Custom Model**:
    - Create a custom CNN architecture with color-based features.
    - Use image augmentation to handle variations in lighting, size, and background.
3. **Baseline Models**:
    - Compare your model's results with pre-trained models like **ResNet** and **DenseNet**.
4. **Goal**:
    - Achieve high classification accuracy on a fruit dataset and provide a performance comparison with SOTA models.

Sample 3: Birds of Thailand Classification

Objective:

Develop a custom model to classify different species of **birds** found in Thailand using images.

Steps:

1. **Dataset**:
    - Collect images of various birds found in Thailand (e.g., **Kingfisher**, **Hornbill**, **Eagle**, **Parrot**).
2. **Custom Model**:
    - Create a custom CNN-based architecture with multiple convolutional layers.
    - Experiment with pooling and dropout layers for overfitting prevention.
3. **Baseline Models**:
    - Compare your model's accuracy with pre-trained models like **ResNet** and **Vision Transformer**.
4. **Goal**:
    - Train and evaluate the custom model on bird classification images to reach competitive performance.

Sample 4: Local Thai Food Classification

Objective:

Create a deep learning model that classifies different types of **Thai food** dishes using images.

Steps:

1. **Dataset**:
   - Collect images of popular **Thai dishes** like **Pad Thai**, **Som Tum**, **Green Curry**, and **Tom Yum Goong**.
   - Label each image with the corresponding dish name.
2. **Custom Model**:
   - Design a custom CNN with advanced layers like **Batch Normalization** and **Global Average Pooling**.
   - Consider using color-based and texture-based features that are typical in food items.
3. **Baseline Models**:
   - Compare with pre-trained models like **ResNet** or **DenseNet** for performance evaluation.
4. **Goal**:
   - Achieve good classification accuracy on food images and evaluate using metrics such as **precision**, **recall**, and **F1-score**.

Sample 5: Plant Identification for Conservation Projects

Objective:

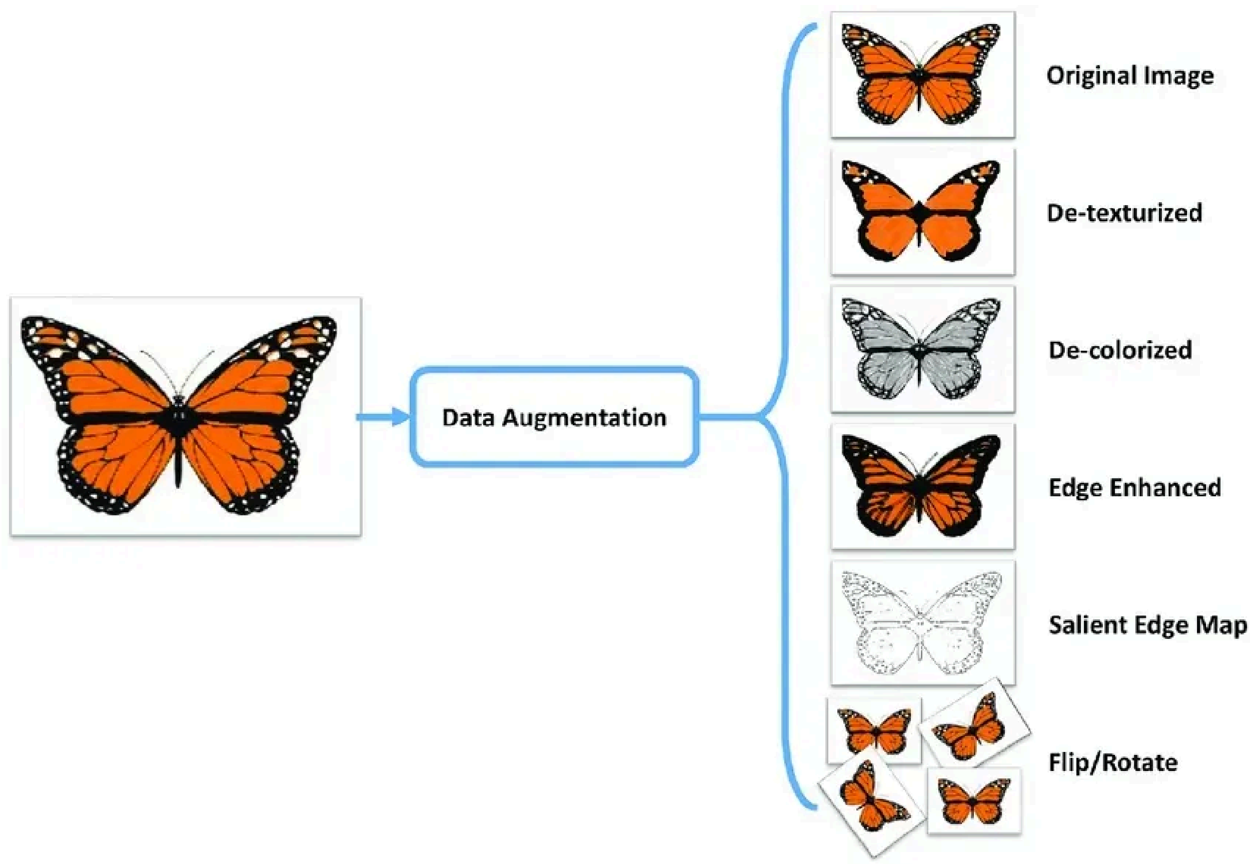
Build a deep learning model to classify **plants** for a **conservation project** (e.g., endangered species, native plants).

Steps:

1. **Dataset**:
    - Capture images of plants from conservation areas, focusing on native and endangered species.
    - Examples include the **Rare Orchid**, the **Golden Bamboo**, and the **Red Lotus**.
2. **Custom Model**:
    - Design a CNN model with additional layers to extract deeper features (e.g., for leaf shapes, petal textures).
    - Use techniques like **Transfer Learning** for fine-tuning if needed.
3. **Baseline Models**:
    - Compare performance with **ResNet**, **DenseNet**, and **Vision Transformer** (ViT).
4. **Goal**:
    - Evaluate the model's ability to correctly identify plant species and provide insights into classification performance for conservation efforts.

Key Notes for All Projects:

- **Image Augmentation**: Use techniques like rotation, flipping, and scaling to improve generalization.
- **Hyperparameter Tuning**: Try different learning rates, batch sizes, and optimizers to find the best configuration.
- **Evaluation Metrics**: Focus on metrics such as **accuracy**, **precision**, **recall**, and **F1-score**.
- **Model Comparison**: After training your custom architecture, compare it against SOTA models like **ResNet**, **DenseNet**, and **Vision Transformer** to see how it performs in a real-world scenario.



Credit: https://www.labellerr.com/blog/what-is-data-augmentation-techniques-examples-benefits/