## 🌞 Faulty Solar Panel Classification using ResNet18

This project classifies solar panel images into 6 categories using **deep learning with transfer learning (ResNet18)**. It covers data preparation, model training, performance monitoring, and prediction.

---

## 📌 Workflow Steps

### 1. Setup

- Selects **GPU if available**, otherwise uses CPU.
- Fixes **random seeds** for reproducibility.
- Defines **number of classes** (6) and training settings.

---

### 2. Data Preparation

- Loads solar panel dataset from a folder with subfolders per class.
- Applies **data augmentation** for training (resize, flips, rotations, normalization).
- Applies **standard preprocessing** for validation (resize, normalization).
- Splits dataset into **80% training and 20% validation**.
- Organizes data into mini-batches for efficient training.

---

### 3. Model Setup

- Uses a **pretrained ResNet18** model.
- Replaces the final layer so it predicts **6 classes** instead of ImageNet classes.
- Chooses **cross-entropy loss** for classification.
- Optimizes with **Adam** optimizer.
- Uses **mixed precision training** to speed up computation on GPU.

---

### 4. Training & Validation

- Runs multiple epochs of training with forward and backward passes.
- Tracks **training loss** after each epoch.
- Validates the model after each epoch, measuring **validation loss and accuracy**.
- Uses **early stopping**: if validation loss doesn’t improve for 2 epochs, training stops.
- Saves the **best model** when validation loss is lowest.

---

### 5. Performance Monitoring

- Stores training and validation loss values, as well as validation accuracy.
- Plots **training vs validation loss** curves.
- Plots **validation accuracy over epochs** to visualize performance.

---

### 6. Inference (Prediction)

- Loads the **best saved model**.
- Preprocesses a new image in the same way as validation data.
- Runs the image through the model to predict the class.
- Displays the image along with its **predicted label**.
- Prints prediction results and **time taken for inference**.

---

### 🚀 Key Highlights

- **Transfer Learning**: Faster convergence using ResNet18 pretrained on ImageNet.
- **Data Augmentation**: Improves generalization by introducing image variations.
- **Early Stopping**: Prevents overfitting by monitoring validation loss.
- **Visualization**: Clear plots for training and validation metrics.
- **User-Friendly Inference**: Quick prediction on any given solar panel image.