End-to-End Deep Learning Workflow

Saeid Nahavandi
Distinguished Professor
snahavandi@swin.edu.au

Parham Kebria
Senior Research Scientist
parhamkebria@ieee.org

Advanced Study Institute: Artificial Intelligence for Disaster Management
Orlando, November 17 – 25, 2025

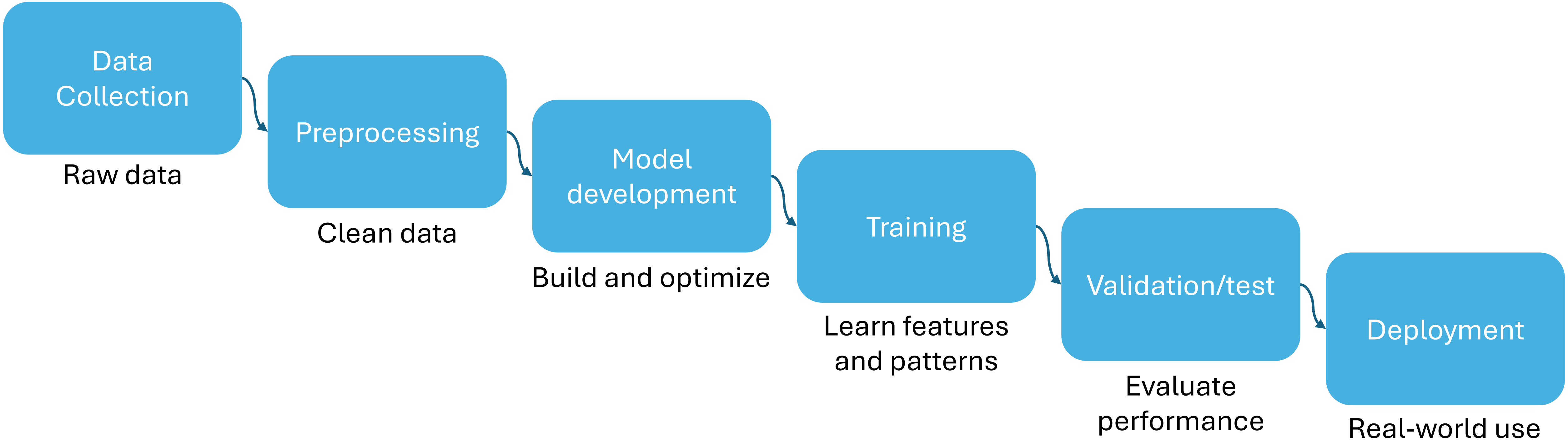
NATO OTAN
This activity is supported by: The NATO Science for Peace and Security Programme

SWINBURNE UNIVERSITY OF TECHNOLOGY
UCF

End-to-End Deep Learning Workflow

Overview of the Pipeline

What is a Deep Learning Workflow?



End-to-End Deep Learning Workflow

Why an End-to-End Pipeline Matters

- Real-world data is messy, not like curated benchmarks
- Most project times spent on *data* and *validation*
- Deployment ensures the model's value is realized

An End-to-End example:

Drone detecting flood-affected areas
→ insights delivered via mobile dashboard
→ response actions to be taken based on the insights

End-to-End Deep Learning Workflow

→ Data Sources Identified & Collected

End-to-End Deep Learning Workflow

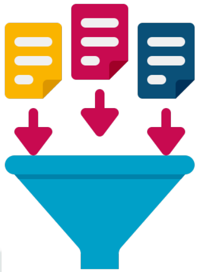
→ Data Sources Identified & Collected

→ Data Preprocessing & Augmentation for Training

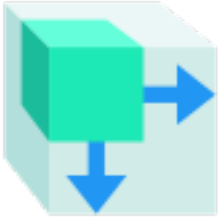
End-to-End Deep Learning Workflow

→ Data Sources Identified & Collected

→ Data Preprocessing & Augmentation for Training



→ Design & Develop the Model Architecture

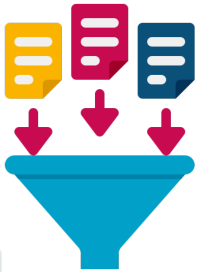


End-to-End Deep Learning Workflow

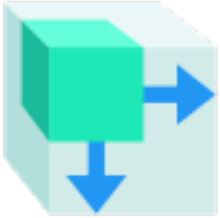
Overview of the Pipeline

⟶ Data Sources Identified & Collected

⟶ Data Preprocessing & Augmentation for Training



⟶ Design & Develop the Model Architecture



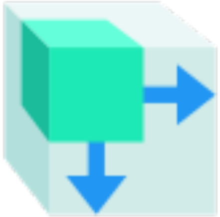
⟶ Model Training & Validation → Model Optimization

End-to-End Deep Learning Workflow

→ Data Sources Identified & Collected

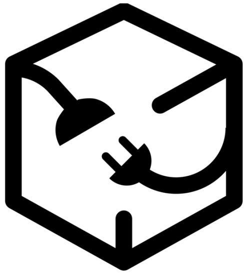
→ Data Preprocessing & Augmentation for Training

→ Design & Develop the Model Architecture



→ Model Training & Validation → Model Optimization

→ Deployment of the Final Model in Practice



End-to-End Deep Learning Workflow

Sources:

- Sentinel-2, Landsat, Planet Labs, Kaggle, etc.

End-to-End Deep Learning Workflow

Sources:

- Sentinel-2, Landsat, Planet Labs, Kaggle, etc.

Challenges:

- Cloud cover, varying lighting, perspective distortion
- Different resolutions and noisy/incorrect labels

End-to-End Deep Learning Workflow

Sources:

- Sentinel-2, Landsat, Planet Labs, Kaggle, etc.

Challenges:

- Cloud cover, varying lighting, perspective distortion
- Different resolutions and noisy/incorrect labels

Labeling Strategies:

- Manual annotation (slow but accurate)
- Crowdsourcing (Amazon MTurk, LabelBox)
- Semi-supervised or weakly labeled approaches

End-to-End Deep Learning Workflow

Sources:
- Sentinel-2, Landsat, Planet Labs, Kaggle, etc.

Challenges:
- Cloud cover, varying lighting, perspective distortion
- Different resolutions and noisy/incorrect labels

Labeling Strategies:
- Manual annotation (slow but accurate)
- Crowdsourcing (Amazon MTurk, LabelBox)
- Semi-supervised or weakly labeled approaches

Tip:
Always inspect a subset of your data visually

End-to-End Deep Learning Workflow

Sources:
- Sentinel-2, Landsat, Planet Labs, Kaggle, etc.

Challenges:
- Cloud cover, varying lighting, perspective distortion
- Different resolutions and noisy/incorrect labels

Labeling Strategies:
- Manual annotation (slow but accurate)
- Crowdsourcing (Amazon MTurk, LabelBox)
- Semi-supervised or weakly labeled approaches

Tip:
Always inspect a subset of your data visually

Handling Noisy Labels

Problem:
- Mislabeling
- Unclear boundaries
- Class imbalance

Solution:
- Cross-check with multiple annotators
- Use polygon-based labeling
- Oversampling/class weights

End-to-End Deep Learning Workflow

Preprocessing Essentials:

- Resize all images to a common input shape
- Normalize pixel values ([0,1] or [-1,1])
- Split dataset → Train / Val / Test

Common Augmentations:

Technique	*Example*	*Purpose*
Flip/Rotate	90°, 180°	Orientation invariance
Random Crop	Random patch	Simulate zooming
Color Jitter	Vary brightness/contrast	Lighting robustness
Gaussian Noise	Add small noise	Robustness to sensor noise

End-to-End Deep Learning Workflow


```python
import torchvision.transforms as T

transform = T.Compose([
    T.Resize((224, 224)),
    T.ToTensor(),
    T.Normalize(mean=[0.485, 0.456, 0.406],
                std=[0.229, 0.224, 0.225])])
```

```python
from tensorflow.keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=15,
    horizontal_flip=True,
    zoom_range=0.1)
```

End-to-End Deep Learning Workflow

End-to-End Deep Learning Workflow

Deployment Options:

Platform	*Use Case*	*Tools*
Mobile App	On-device detection	TensorFlow Lite / Core ML
Drone	Real-time edge inference	NVIDIA Jetson / ONNX
Web API	REST-based model serving	FastAPI / Flask + TorchServe

End-to-End Deep Learning Workflow

🎯 ***This Session:***

- ❖ Deep learning = pipeline, not just model
- ❖ Data quality >> Model complexity
- ❖ Always visualize and validate at each step
- ❖ Deployment ensures real-world impact

📎 ***Previous sessions***:

- ✓ Image Classification Fundamentals
- ✓ CNN Architectures
- ✓ Transfer Learning & Fine-Tuning
- ✓ Full pipeline integration

End-to-End Deep Learning Workflow

Quiz!

1. What is the most time-consuming stage of a deep learning project?
   a) Training
   b) Data collection and cleaning
   c) Deployment

2. Why is normalization important before training?
   a) Makes training faster and more stable
   b) Adds noise
   c) Reduces dataset size

3. What is data augmentation used for?
   a) Reduce accuracy
   b) Improve generalization
   c) Label correction

End-to-End Deep Learning Workflow

Quiz!

1. What is the most time-consuming stage of a deep learning project?
   - a) Training
   - ✅ b) Data collection and cleaning
   - c) Deployment

2. Why is normalization important before training?
   - ✅ a) Makes training faster and more stable
   - b) Adds noise
   - c) Reduces dataset size

3. What is data augmentation used for?
   - a) Reduce accuracy
   - ✅ b) Improve generalization
   - c) Label correction

End-to-End Deep Learning Workflow

(Q&A)

Parham Kebria
parhamkebria@ieee.org