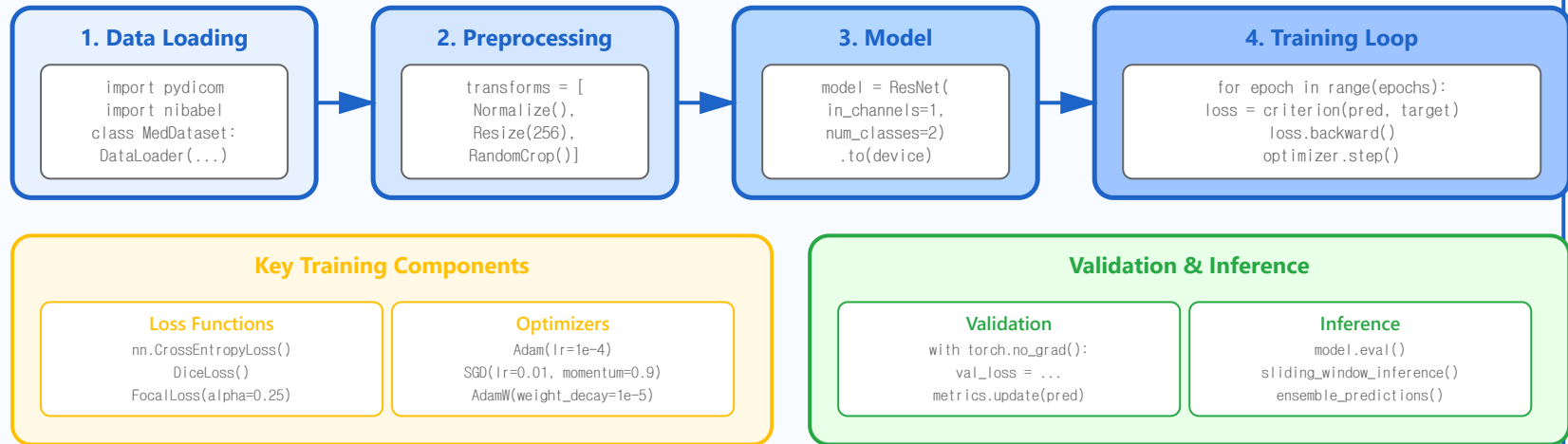# Hands-on: PyTorch Medical Imaging

Complete PyTorch Pipeline for Medical Imaging

1. Data Loading

```
import pydicom
import nibabel
class MedDataset:
    DataLoader(...)
```

2. Preprocessing

```
transforms = [
    Normalize(),
    Resize(256),
    RandomCrop()]
```

3. Model

```
model = ResNet(
    in_channels=1,
    num_classes=2)
    .to(device)
```

4. Training Loop

```
for epoch in range(epochs):
    loss = criterion(pred, target)
    loss.backward()
    optimizer.step()
```

Key Training Components

Loss Functions
```
nn.CrossEntropyLoss()
DiceLoss()
FocalLoss(alpha=0.25)
```

Optimizers
```
Adam(lr=1e-4)
SGD(lr=0.01, momentum=0.9)
AdamW(weight_decay=1e-5)
```

Validation & Inference

Validation
```
with torch.no_grad():
    val_loss = ...
    metrics.update(pred)
```

Inference
```
model.eval()
sliding_window_inference()
ensemble_predictions()
```

Data Loading

DICOM reading with pydicom, NIfTI with nibabel. Custom Dataset classes

Model Implementation

ResNet, U-Net from scratch or torchvision. Custom layers for medical imaging

Training Loops

Loss functions, optimizers, learning rate schedules. Mixed precision training

Validation

Metrics computation, checkpointing. Early stopping and model selection

Inference

Sliding window for large images. Batch processing and result aggregation