MSc Research Practicum

CONFIGURATION MANUAL

Enhancing Leukemia Diagnosis with Synthetic Data and Explainable Deep Learning Architectures

x23373385
Ibrahim Malik
MSc Data Analytics
Group B

This configuration manual provides all necessary parameters and settings to replicate the experimental setup for the leukemia diagnosis research project.

Table of Contents

1. Environment Requirements

Platform Setup

- **Platform**: Google Colab Pro (required for enhanced GPU/TPU access)
- **Python Version**: 3.12
- **Storage**: Google Drive integration for dataset access
- **Hardware**: GPU runtime (Tesla T4)

Required Libraries

All libraries use latest available versions:

- TensorFlow/Keras
- NumPy
- OpenCV-Python
- Matplotlib
- Scikit-learn
- SHAP
- LIME
- Scipy

Installation Commands

```
pip install tensorflow keras numpy opencv-python matplotlib scikit-learn shap lime scipy
```

2. Dataset Configuration

Dataset Source

- **Primary Source**: The Cancer Imaging Archive (TCIA)
- **URL**: C_NMC_2019 Dataset: ALL Challenge dataset of ISBI 2019 (C-NMC 2019) - The Cancer Imaging Archive (TCIA) Public Access - Cancer Imaging Archive Wiki
- **Dataset Name**: C-NMC (Children's Hospital of Philadelphia)

Directory Structure

```
/content/drive/MyDrive/leukemia_detection_project/
├── C-NMC/
│   ├── training_data/
│   │   ├── fold_0/
│   │   │   ├── all/      (2397 cancer images)
│   │   │   └── hem/      (1130 normal images)
│   │   ├── fold_1/
│   │   │   ├── all/      (2418 cancer images)
│   │   │   └── hem/      (1163 normal images)
│   │   └── fold_2/
│   │       ├── all/      (2457 cancer images)
│   │       └── hem/      (1096 normal images)
│   └── validation_data/
│       ├── all/          (1219 cancer images)
│       └── hem/          (648 normal images)
└── Code/
    ├── GANs/
    ├── CNNs/
    ├── ViTs/
    ├── hybrid_model_integration/
    └── XAI/
```

Image Preprocessing

- **Target Size**: 224×224 pixels
- **Normalization**: [0,1] scale (divide by 255.0)
- **Color Mode**: RGB (3 channels)
- **Data Type**: float32

Data Augmentation (Training)

- Rotation: ±20 degrees
- Width/Height Shift: ±10%
- Shear: ±10%
- Zoom: ±10%
- Horizontal Flip: Enabled
- Rescale: 1/255

3. Model Configurations

3.1 GAN Architectures

DCGAN Configuration

- **Training**: 225 epochs, batch_size=32
- **Image Normalization**: [-1,1] range
- **Latent Dimension**: 100
- **Generator**: 7 Conv2DTranspose layers (7×7->224×224, tanh output)
- **Discriminator**: 2 convolutional layers (stride=2, sigmoid output)
- **Optimizers**:
    - Discriminator: Adam(lr=0.0002, β_1=0.5, clipvalue=1.5)
    - Generator: Adam(lr=0.0001, β_1=0.5)
- **Loss**: Binary cross-entropy

WGAN Configuration

- **Training**: 200 epochs, batch_size=32
- **Latent Dimension**: 500
- **Training Protocol**: 5 critic iterations per generator update
- **Gradient Penalty**: λgp=10
- **Generator**: Dense->7×7, then 5 Conv2DTranspose layers (tanh output)
- **Critic**: Identical to DCGAN discriminator (no sigmoid)
- **Optimizer**: Adam(lr=0.0001, β_1=0.5) for both networks

cGAN Configuration

- **Training**: 250 epochs, batch_size=32
- **Latent Dimension**: 500
- **Classes**: 0=healthy, 1=leukemic
- **Generator**: Dense->7×7, then 5 Conv2DTranspose + concatenated class labels
- **Discriminator**: Receives both image and label inputs
- **Optimizers**:
    - Generator: Adam(lr=0.0002, β_1=0.5)
    - Discriminator: Adam(lr=0.00005, β_1=0.5, clipvalue=1.0)

3.2 CNN Architecture

- **Input Shape**: (224, 224, 3)
- **Architecture**:
    1. Conv2D(32, 3×3) + ReLU -> MaxPooling2D(2×2)
    2. Conv2D(64, 3×3) + ReLU -> MaxPooling2D(2×2)
    3. Conv2D(128, 3×3) + ReLU -> MaxPooling2D(2×2)
    4. Flatten -> BatchNormalization
    5. Dense(128) + ReLU -> Dropout(0.5)
    6. Dense(1) + Sigmoid
- **Optimizer**: Adam(lr=0.0001)
- **Loss**: Binary cross-entropy
- **Training**: 50 epochs, batch_size=32

3.3 Vision Transformer (ViT)

- **Input Shape**: (224, 224, 3)
- **Patch Size**: 16×16 (P=16)
- **Number of Patches**: 196 (N=(224/16)²)
- **Embedding Dimension**: 256 (D=256)
- **Transformer Layers**: 4 (L=4)
- **Attention Heads**: 4 (h=4)
- **Head Dimension**: 64 (dk=64)
- **MLP Hidden Sizes**: [512, 256]
- **Activation**: GELU
- **Optimizer**: Adam(lr=1e-4)
- **Loss**: Categorical cross-entropy
- **Training**: 30 epochs, batch_size=32

3.4 Hybrid CNN-ViT

- **CNN Branch**: 4 conv blocks (filters: 32, 64, 128, 256)
- **ViT Branch**: Processes CNN feature maps, 16×16 patches -> 9 patches
- **Patch Dimensions**: 65,536 (16×16×256)
- **Transformer Blocks**: 6 (L=6)
- **Fusion Method**: Element-wise addition
- **Classification Head**: [512, 256] hidden units, GELU activation
- **Training**: 50 epochs, batch_size=32, Adam(lr=1e-4)

4. XAI Configuration

4.1 Grad-CAM Settings

- **Target Layers**:
  - CNN: 'conv2d_2' (last convolutional layer)
  - Hybrid: 'conv2d_3'
- **Class Indices**: 0=Healthy, 1=Leukemia
- **Visualization**: Alpha=0.6, Colormap='jet'
- **Quality Threshold**: heatmap_std > 0.1

4.2 SHAP Configuration

- **Explainer Type**: Partition Explainer (primary), Deep Explainer (fallback)
- **Parameters**:
  - num_samples=50 (background images)
  - max_evals=500 (model evaluations)
  - batch_size=32
- **Masker**: Inpainting (preferred), blur(1,1) (fallback)

4.3 LIME Settings

- **Parameters**:
  - num_samples=1000 (perturbed samples)
  - num_features=5 (top superpixels to highlight)
  - positive_only=True
  - hide_rest=False
- **Segmentation**: quickshift(kernel_size=4, max_dist=200, ratio=0.2)

5. File Organization

Code Structure

```
Code/
├── GANs/
│   ├── saved_models/
│   │   ├── dcgan/ (dcgan_generator.keras, dcgan_discriminator.keras)
│   │   ├── wgan/ (wgan_generator.keras, wgan_discriminator.keras)
│   │   └── cgan/ (cgan_generator.keras, dcgan_discriminator.keras)
│   ├── dcgan.ipynb
│   ├── wgan.ipynb
│   ├── cgan.ipynb
│   ├── gan_image_generator.ipynb
│   └── gan_evaluation.ipynb
├── CNNs/
│   ├── unbalanced-cnn/ (unbalanced_cnn.ipynb, cnn_unbalanced_model.keras)
│   ├── dcgan-cnn/ (dcgan_cnn.ipynb, cnn_dcgan-balanced_model.keras)
│   ├── wgan-cnn/ (wgan_cnn.ipynb, cnn_wgan-balanced_model.keras)
│   └── cgan-cnn/ (cgan_cnn.ipynb, cnn_cgan-balanced_model.keras)
├── ViTs/
│   ├── unbalanced-vit/ (unbalanced_vit.ipynb, vit_unbalanced_model.keras)
│   ├── dcgan-vit/ (dcgan_vit.ipynb, vit_dcgan-balanced_model.keras)
│   ├── wgan-vit/ (wgan_vit.ipynb, vit_wgan-balanced_model.keras)
│   └── cgan-vit/ (cgan_vit.ipynb, vit_cgan-balanced_model.keras)
├── hybrid_model_integration/
│   ├── cnn-vit-hybrid.ipynb
│   ├── dcgan-cnn-vit-hybrid.ipynb
│   ├── wgan-cnn-vit-hybrid.ipynb
│   └── cgan-cnn-vit-hybrid.ipynb
└── XAI/
    ├── Grad-CAM.ipynb
    ├── SHAP.ipynb
    └── LIME.ipynb
```

6. Training Configuration

Callbacks (All Models)

- **Early Stopping**: monitor='val_loss', patience=5
- **Model Checkpoint**: save_best_only=True
- **ReduceLROnPlateau**: factor=0.2, patience=3, min_lr=1e-7

Evaluation Metrics

- Accuracy, Precision, Recall, F1-Score, AUC-ROC
- Sensitivity, Specificity
- FID, SSIM, Inception Score (for GANs)

7. Common Issues & Solutions

Memory Issues

- **Symptoms**: ResourceExhaustedError, CUDA out of memory
- **Solutions**: Reduce batch_size to 16 or 8, enable GPU memory growth

Training Issues

- **Symptoms**: Model not converging, NaN values
- **Solutions**: Check learning rates (try 1e-4 or 1e-5), verify data normalization

GAN Issues

- **Symptoms**: Mode collapse, training instability
- **Solutions**: Balance G/D training rates, use label smoothing, try WGAN for stability

Colab Issues

- **Symptoms**: Session timeouts, runtime disconnections
- **Solutions**: Use Google Colab Pro, save checkpoints frequently, remount Drive if needed