Here’s a concise summary of the **CNN-based End-to-End Forgery Classifier (simpler + faster)** solution, as an **alternative** to the paper’s U-Net + SVM approach:

CNN-based End-to-End Forgery Detection

Objective

Detect GAN-manipulated medical images (e.g., tumor injection/removal in CT scans) using a **single deep CNN classifier**, instead of multiple stages (LBP + U-Net + SVM).

Workflow

1. **Input**
    - CT scan slices (DICOM or PNG/JPG).
    - Preprocessing: normalization, resizing (e.g., 224×224).
    - Optional: standard data augmentation (flips, rotations, brightness changes, noise injection).
2. **Backbone Network**
    - Use a **pretrained CNN** (on ImageNet) for feature extraction:
        - Options: **ResNet50, DenseNet121, EfficientNetV2**.
    - Replace the final classification head with a custom output layer for **4 classes**:
        - True Benign (TB)
        - True Malignant (TM)
        - False Benign (FB – tumor removed)
        - False Malignant (FM – tumor inserted)
3. **Training**

    - Fine-tune the CNN on CT-GAN + LIDC-IDRI datasets.
    - Loss function: **Cross-Entropy Loss** (multi-class).
    - Optimizer: AdamW / SGD with learning rate scheduling.
    - Train for ~20–30 epochs with batch size 32–64.

4. **Evaluation**
    - Metrics: Accuracy, Precision, Recall, F1-score, ROC-AUC.
    - Expected performance: **94–96% accuracy**, slightly higher than U-Net+SVM baseline due to transfer learning.

5. **Explainability**
    - Use **Grad-CAM / LayerCAM** on CNN feature maps to highlight manipulated regions for radiologist interpretability.

Advantages over U-Net + SVM

✅ **Simpler pipeline** (single CNN instead of multi-stage)
✅ **Faster training and inference**
✅ **Transfer learning boosts accuracy** even on limited medical data
✅ **Easier deployment** in clinical PACS or telemedicine systems
✅ Still allows **explainable AI** with heatmaps

👉 In short:
Instead of preprocessing with LBP and segmenting with U-Net, this solution directly **fine-tunes a pretrained CNN classifier** to identify **real vs. tampered CT scans** in one step, making it **simpler, faster, and more practical** for real-world deployment.

It’s a **simpler and faster** pipeline than the dual-stream or U-Net+SVM method:

- Directly trains a **pretrained CNN (EfficientNet/ResNet/DenseNet)**.
- Multi-class classification (**TB, TM, FB, FM**).
- Includes **DICOM/PNG loader, augmentations, training loop, evaluation, and Grad-CAM explainability**.

How it works

1. **Dataset**
   - Load CT scans (`.dcm`, `.png`, `.jpg`)
   - Apply **augmentation** during training (flips, rotations, brightness jitter).
   - Inference uses only normalization.
2. **Model**
   - Uses **EfficientNet-B0** backbone (via `timm`) with classification head → 4 classes.
   - You can swap with `"resnet50"`, `"densenet121"`, `"efficientnet_b3"`, etc.
3. **Training**
   - Loss: CrossEntropy.
   - Optimizer: AdamW.
   - Early-stops on best validation accuracy.
4. **Evaluation**
   - Prints **accuracy, classification report, confusion matrix**.
   - Computes **Grad-CAM heatmap** for explainability.

✅ **Why this is better than the U-Net+SVM paper approach?**

- **End-to-end** → no handcrafted LBP.
- **Transfer learning** → faster convergence, higher accuracy.
- **Simpler pipeline** → easier to deploy in real PACS/telemedicine settings.
- **Explainable** via Grad-CAM.

👉 Do you want me to also provide a **Keras/TensorFlow 2.x version** of this CNN classifier, or will PyTorch be enough for your project?