Step	Description	Original Implementation	Updates/Improvements	Why It Helps	Notes
Resampling	Convert all audio to a fixed sampling rate	LJSpeech audio used original 22.05 kHz	All audio explicitly resampled to 22.05 kHz	Ensures consistent audio input size and features for spectrogram computation	Required for both real and fake samples
Silence Trimming	Remove leading and trailing silence	Not always applied	Applied using librosa.effects.split	Reduces irrelevant silent parts, improves model learning efficiency	Threshold top_db=40
LUFS Normalization	Loudness normalization	Previously done roughly	Added pyloudnorm with target -23 LUFS, clipping handled via np.clip	Ensures consistent perceived volume, prevents extreme amplitude values	Prevents over-amplified samples from biasing the model
Clipping Fix	Prevent audio exceeding [-1, 1] range	Not explicitly clipped	Explicit np.clip(audio, -1.0, 1.0)	Avoids distortion and warnings for clipping	Still generated minor warnings (~17 in dataset)
Mel-Spectrogram Computation	Convert audio to spectrogram	Original computation via librosa.feature.melspectrogram	Same, but standardized and resized consistently	Converts temporal audio signal into 2D visual-like representation for CNN input	Parameters: n_mels=80, n_fft=2048, hop_length=512
Log Scaling	Power-to-dB scaling	Used librosa.power_to_db	Retained	Makes spectrogram values perceptually linear	Standard practice for audio deep learning
Standardization	Normalize spectrogram	Previously done only via min-max normalization	Standardization applied: (mel - mean)/std	Stabilizes CNN training, improves gradient flow	Avoids vanishing/exploding features
3-Channel Stacking	Convert spectrogra	Not originally applied	Stack grayscale spectrogram 3 times	Enables direct use	CNNs like ResNet pre-

Step	Description	Original Implementation	Updates/Improvements	Why It Helps	Notes
	m to 3 channels		to create 3-channel image	with pre-trained CNNs (ResNet) expecting 3 channels	trained on ImageNet require 3 channels
Resize Spectrograms	Uniform image size	Originally 299x299	Maintained 299x299 for LJSpeech; optionally downscaled 128x128 for WaveFake	Reduces memory footprint, ensures consistent input shape	For Colab Free GPU, smaller size recommended for large datasets
PNG Generation	Save spectrogram images	Saved spectrograms without axes	Updated to include axes, labels, colorbar	Visual inspection, debugging, model interpretability	Optional for WaveFake due to memory constraints
Train/Test/Val Split	Split datasets	LJSpeech split not clear	Explicit 80/10/10 split for LJSpeech; WaveFake splitting deferred	Ensures proper evaluation	Splits contain subfolders for audio, mel, images
Audio Saving	Save normalized audio	Used deprecated librosa.output.write_wav	Switched to soundfile.write (sf.write)	Compatible with modern Python and avoids warnings	Audio files remain normalized, clipped within [-1, 1]
Logs	Track preprocessing	No logs originally	Added logs folder to record warnings (e.g., clipped samples)	Helps identify issues, reproducibility	CSV file optionally tracks all filenames, splits, vocoders
WaveFake Handling	Process synthetic audio	Previously partial updates	Full preprocessing with 8 vocoder folders, skipping Japanese-only folders	Ensures real vs fake classification consistency	PNG generation skipped for memory, LUFS normalization applied
Dataset Merging	Real + fake	Not done originally	Merged later for classifier training, balanced real/fake	Ensures fair training, reduces bias	Balancing done using augmentation on real samples during training