AI Image/Video Deepfake Detection - Roadmap

1. Why a Vision Model, Not Just an LLM

- LLMs process text, they do not see pixels.
- Vision or multimodal encoders (CNN, ViT, CLIP) are needed to spot GAN artefacts.
- Plan: use a large vision model for training, then distil/quantise for mobile.

2. Two-Tier Real-Time Architecture

React Native App
  * Thumbnail (224x224) pre-filter - tiny CNN (<10 ms)
  * Sends suspicious frames only to heavy model

Local / Cloud Inference API
  * FastAPI, loads larger ONNX model
  * Returns JSON verdict + confidence

3. Model Building Phases

Phase 0 - Faces only
  Dataset: 140k Real vs Fake Faces
  Model: MobileNet-small (5 MB) -> on-device

Phase 1 - Full images
  Dataset: RealStock + Diffusion outputs
  Model: EfficientNet-B0 (14 MB) -> local API

Phase 2 - Video
  Dataset: FaceForensics++, DFDC
  Model: MesoNet-LSTM (25 MB) -> local API

Phase 3 - Multimodal
  Dataset: LAION AI-Fake mixed set
  Model: Distilled CLIP tiny (30 MB) -> GPU cloud

4. Training Pipeline (Local)
---
datasets/
  real/   (JPEGs)
  fake/   (GAN images)

$ python train/train_model.py --arch mobilenet_v2 --epochs 5 --out models/mobilenet_s.onnx

Export to ONNX:
torch.onnx.export(model, dummy_input, "models/model.onnx", ...)

5. Integration Checklist

[ ] Gather 1k real + 1k fake images
[ ] Train Phase-0 model, export ONNX
[ ] Add thumbnail filter in React Native
[ ] Build FastAPI endpoint /detect
[ ] Log predictions for manual review

Later:
[ ] Expand dataset to 50k+
[ ] Train video model
[ ] Distil INT8, benchmark 60 fps scroll
[ ] Migrate API to GPU server

6. Where an LLM Fits Later
---
* Generate human-readable explanations
* Cluster false positives for retraining
* Cross-modal sanity (caption vs image)

End of Roadmap.