AI Image/Video Deepfake Detection - Roadmap

This roadmap outlines how to evolve a prototype mobile detector into a real-time, scroll-time product - suitable for VC review.

1 | Why You Need Vision Models, Not Just LLMs

* LLMs process text; they do not understand pixels.
* GAN and diffusion artefacts are visual - use CNN / ViT / CLIP encoders.
* Plan: train large vision or multimodal models, then distil / quantise for edge devices.

2 | Two-Tier Real-Time Architecture

React Native App
  * 224x224 on-device thumbnail filter (tiny CNN, <10 ms)
  * Only sends high-suspect media 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 filter

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/   - photographs
  fake/   - GAN or diffusion
```

```
$ 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 | On-Device Pre-Filter - Detailed Explanation

Purpose

* Acts as a bouncer: instant 224x224 scan per scroll item.
* Keeps UX smooth; saves battery and bandwidth.

Workflow

1. Resize frame to 224x224.
2. Tiny CNN outputs fake-probability p.
   - p < 0.3 -> mark safe.
   - p > 0.7 -> send full-res to heavy model.
3. Heavy model corrects any false alarms.

Model Choices

* MobileNet-V2 tiny (3 MB, 5-8 ms)
* ShuffleNet-V2 0.5x (2.5 MB, 4-6 ms)
* MesoNet-lite (1 MB, ~3 ms)

Training Tips

* Use same dataset but resized and heavily augmented.
* Stop at ~85% accuracy; heavy model is safety net.
* Quantise to INT8 for TFLite / ONNX.

Threshold Tuning

* Safe cut-off: false positive rate <= 3%.
* Suspect cut-off: true positive rate >= 85%.

Performance Proof (Pixel 5)

* 7 ms per frame, 120 thumbnails / s, 30% CPU used.

Limitations & Mitigations

* Might miss subtle fakes - send uncertain frames to heavy model.
* Thumbnail blur - sharpen lightly before inference.

7 | Where an LLM Fits Later

* Generate user-friendly explanations of detected artefacts.
* Cluster false positives for rapid retraining.
* Cross-modal sanity checks: compare Whisper transcript with lip movement.

End of Roadmap

Questions? Contact: your-email@company.com