MVP (90 dias): Forense de Deepfake + Manipulação + Fingerprint de Compressão

Objetivo do MVP (o que "precisa funcionar" em 3 meses)

Um pipeline único, auditável, que pega **imagem ou vídeo** e gera um **Relatório Técnico** com:

1. **Aquisição & preservação**

- hash SHA-256 do arquivo
- metadados técnicos (codec, dimensões, subsampling, progressive/baseline, EXIF/ICC)

2. **Atribuição por compressão (JPEG Quantization Fingerprint)**

- extrai DQT (quant tables)
- compara com seu banco "Photoshop/GIMP/Imagemagick etc."
- retorna *compatibilidades prováveis* (não "prova absoluta")
  👉 Isso tem base científica sólida desde Kornblum 2008.

3. **Detecção de deepfake (imagem/vídeo)**

- frame sampling + face crop
- detector baseline treinado/ajustado em dataset padrão
- score + calibração + taxa de erro documentada
  DeepfakeBench é um bom "esqueleto" aqui: ele organiza métodos e datasets de deepfake detection.

4. **Localização de manipulação em imagem (heatmap)**

- rodar Comprint (compressão) e, quando possível, fusão com Noiseprint
  Comprint foi desenhado pra "in-the-wild" e é treinado com **pristine data only** (um detalhe ouro).

5. **Saída pericial pronta**

- JSON + PDF/HTML com evidências, hashes, parâmetros, versões, commits
- linguagem pericial: "compatível com...", "consistente com...", "limitações..."

Dataset "completo" (o que vocês precisam juntar)

Aqui é onde se ganha ou se perde o projeto.

Camada A — Banco de Quantização (seu projeto: quantization_extend)

Finalidade: atribuição do "último encoder provável" e estimativa de "quality family".

Dataset mínimo (MVP):

- Photoshop:
  - Save As (baseline)
  - Export As / Save for Web (baseline)
- GIMP:
  - baseline + progressive
- ImageMagick/libjpeg-turbo (opcional, mas muito útil)

Variáveis controladas (obrigatórias):

- Quality: 1..100
- Subsampling: 4:4:4 e 4:2:0 (as duas mais comuns)
- Progressive: sim/não
  📌 Se você não controla isso, você cria uma base "misturada" e ela fica fraca em tribunal.

✅ **Boa prática**: cada amostra precisa ter:

- `software_family`, `export_mode`, `quality`, `subsampling`, `progressive`
- `qtable_Y`, `qtable_C`
- `qhash_Y`, `qhash_C`
- `sha256` do arquivo

Camada B — Deepfake (vídeo/face swap/reenactment)

Finalidade: treinar/validar o detector de deepfake.

Pra não virar um projeto infinito, seu MVP deve focar em **2 datasets grandes** + 1 dataset "difícil":

- **FaceForensics++ (FF++)** (clássico e didático)
- **Celeb-DF v2** (mais "in the wild")
- **DFDC subset** (se você tiver storage/tempo)

O DeepfakeBench foi criado justamente pra padronizar isso e já cita múltiplos datasets integrados.

📌 **Erro de amador aqui:** split errado por identidade
Se você treina com o mesmo rosto aparecendo no treino e no teste, você infla seu resultado e toma pancada no contraditório.

✅ Regra pericial: split por **identidade** e por **vídeo origem**.

Camada C — Manipulação de imagem "não-deepfake" (splicing/copy-move)

Finalidade: localizar adulteração por heatmap e reforçar laudo.

O Comprint é forte aqui e foi avaliado em datasets de manipulação variados; ele existe justamente pra isso e trabalha com inconsistências de compressão.

Você pode aproveitar também listas de datasets consolidadas (pra escolher os melhores com GT mask).

Escolha 2 datasets com máscara (MVP):

- um voltado a splicing
- um voltado a copy-move
- idealmente um com recompressão realista

O que dá pra aproveitar HOJE dos repositórios que seu colega mandou

✅ 1) GRIP/UNINA (como referência e mapa do território)

O GRIP tem linha forte de forense multimídia e é coautor no Comprint.
Use como:

- referência no laudo (fundamentação técnica)
- seleção de datasets e papers "de base"

✅ 2) Comprint (IDLabMedia)

Aproveitamento imediato:

- baseline de **localização** (heatmap)
- metodologia de "pristine-only training", que é pericialmente elegante (menos "viciado em fake específico")

MVP realista:

- você não precisa "reinventar Comprint"; você integra como módulo e mede desempenho no seu laboratório.

✅ 3) DeepfakeBench

Aproveitamento imediato:

- estrutura de treino/avaliação e suporte a múltiplos datasets/métodos

- te dá métricas, protocolos e “chão” pra comparar.

Plano de 3 meses (com divisão de tarefas)

Vou assumir dois papéis:

- **Você (Perito A)**: dono do laudo, cadeia de custódia, reprodutibilidade, fingerprint JPEG
- **Colega (Perito B)**: datasets, automações, deepfake training/eval, integração Comprint

MÊS 1 — Fundamentos que evitam desastre

Semana 1: laboratório “à prova de amador”

Perito A

- define estrutura de pastas e naming
- define schema do DB (JSON/CSV/SQLite)
- define hashing + manifesto de evidência (por arquivo)

Perito B

- baixa e organiza 2 datasets deepfake (mínimo)
- define split por identidade
- prepara pipeline de frames + face-crop

✅ Entregável: “Laboratório pronto”

- `/datasets/` com manifest
- `/experiments/` versionado
- `/reports/` com template de laudo

Semana 2: Dataset de quantização (começar certo)

Perito A

- gera JPEG 1..100 do Photoshop em 2 modos (Save As + Export)
- controla subsampling/progressive
- extrai qtables + qhash + sha256

Perito B

- gera GIMP 1..100 baseline/progressive
- documenta versão do software + flags exatas

✅ Entregável: **Banco de quantização v0.1**

- mínimo: Photoshop (2 modos) + GIMP (2 modos)

Semanas 3–4: ingestão e validação (sem treinar ainda)

Perito A

- implementa verificação automática:
  - JPEG meta (progressive/subsampling)
  - duplicatas
  - inconsistência de labels
- cria “match engine” qhash→origem provável

Perito B

- roda baseline do DeepfakeBench em subset pequeno
- roda Comprint em subset pequeno e salva heatmaps

✅ Entregável: pipeline “end-to-end” rodando em 10 amostras reais (com relatório gerado, mesmo que ainda fraco)

MÊS 2 — MVP operacional (treino e métricas)

Semanas 5–6: deepfake detector baseline “bom o suficiente”

Perito B (principal)

- treina baseline com 1–2 métodos (não 15!)
- calibra limiar e mede:
  - AUC
  - FPR/FNR
  - EER (se possível)

Perito A

- formaliza “regra de evidência” pro laudo:
  - quando score vira indicativo vs forte
  - como reportar incerteza

✅ Entregável: detector com métricas reprodutíveis + splits corretos

---

Semanas 7–8: integração Comprint + quantização no relatório

Perito A (principal)

- integra DB de quantização no pipeline final
- gera seção “compatibilidade de compressão”

Perito B

- integra Comprint heatmap no pipeline
- cria rotinas de export (png overlay + json)

✅ Entregável: relatório automático "v1" com 3 blocos:

- compressão (quant)
- deepfake score
- heatmap manipulação

MÊS 3 — Robustez "de tribunal"

Semanas 9–10: testes adversariais (o que derruba amadores)

Rodar bateria:

- WhatsApp recompress
- recorte e re-save
- screenshot de tela
- re-encode com quality diferente

✅ Entregável: tabela "o que quebra / o que sobrevive"
Isso vira ouro na conclusão do laudo ("limitações do método").

Semanas 11–12: empacotamento e governança

Perito A

- checklist ISO 27037 (aquisição, preservação, rastreabilidade)
- template final de laudo (com anexos técnicos)

Perito B

- Docker/venv fixo com versões
- CI simples (rodar um teste de extração e um match)

✅ Entregável: MVP entregável como "produto de laboratório"

- 1 comando → relatório sai

Erros clássicos que vocês vão evitar (porque isso derruba perito no contraditório)

1. **Misturar export modes** (Photoshop Save As vs Export) no mesmo "quality"
2. Ignorar **subsampling/progressive** → base vira "fantasma"
3. Split errado (mesma pessoa no treino e teste) → resultado inflado
4. Dataset contaminado (imagem-fonte não-lossless)
5. Não registrar versões, seeds e hashes (reprodutibilidade zero)
6. Confundir "compatível com Photoshop" com "prova de Photoshop"
7. Não reportar FPR/FNR → score vira "achismo estatístico"

O que vocês conseguem colocar HOJE no laboratório (checklist prático)

✅ Pasta padrão:

- `datasets/deepfake/{ffpp,celebdf,dfdc_subset}/`
- `datasets/forgery/{casia,copy_move,...}/`
- `datasets/quant/{photoshop_saveas,photoshop_export,gimp_baseline,gimp_progressive}/`

✅ Para cada arquivo:

- `sha256`
- `origem/licença`
- `label`
- `split`
- `parâmetros de geração` (no caso do quant)

✅ Ferramentas instaladas:

- pipeline de extração (quantization_extend)
- Comprint (módulo)
- DeepfakeBench (módulo)