Technical Request Brief – Logo Detection Acceleration

■ *Business Context*

The client operates in the sports and entertainment analytics space. A key offering involves detecting brand logos (e.g., Kia, Sprite, Nike) from video footage (e.g., basketball, football broadcasts or social media content) to quantify brand visibility and sponsorship value.

They currently rely on custom-trained classification models for logo detection, but:

- Collecting enough labeled examples (~200+) takes several weeks

- Model training and deployment adds further latency

- Not all new clients justify the time or data investment upfront

They're actively looking for a fast-turnaround solution to start detecting logos with low setup overhead, even if it's not as robust as a full ML pipeline.

■ *Technical Problem to Solve*

Can we design a lightweight logo detection method that:

- Works from a small number of logo templates (reference images)

- Analyzes long videos (2–3 hours) to identify frames where the logo appears

- Handles moderate camera motion (e.g., basketball games with consistent panning)

- Requires little to no fine-tuning per brand or client

This is not a full-blown real-time detection system but more of an intermediate step to:

- Provide quicker value to new clients

- Avoid weeks of manual labeling and model training

■■ *Initial Ideas Discussed*

1. Template Matching (ORB / Feature Matching):

- Already being tested in-house by client

- Challenges: high latency, brittle under camera angle changes

2. Hybrid Approach:

- Use ORB or SIFT matching as fast MVP

- Pre-process video into frame sets based on heuristics

- Combine with shallow feature embedding & vector similarity (e.g., using CLIP / DINOv2)

3. Few-Shot Logo Detection via Vision Transformers:

- Use pretrained models (e.g., Grounding DINO, OWL-ViT) with a few reference images

4. Fast Tracking Post-Match:

- Once a logo appears, track it across frames to reduce redundant processing

■ *Deliverables for First Phase*

- A Jupyter Notebook or script prototype:

* Takes a reference image (logo template)

* Takes a video file or image sequence

* Returns timestamps or frame indices where logo is detected

* Includes basic precision/recall metrics

- Bonus: demo video showing detections on sample NBA footage

■ *Constraints*

- Client is cost-sensitive: no expensive infrastructure or large inference costs

- Solution should be ready to test on real sports videos within 1–2 weeks

- Python is the preferred language

- Existing codebase is in GitHub; we may integrate there later

■ *Next Steps*

- Internal brainstorm on viable approaches (early this week)

- Assign exploration tasks (e.g., test OWL-ViT, benchmark ORB speed)

- Philippe to deliver consolidated proposal and working prototype by [target date]