[카카오 AI 스타트업 육성프로젝트] 창업 제안서

Contact: limsihyun@kaist.ac.kr or kjh0209@kaist.ac.kr
작성: KAIST 기술경영학부 김지혁, KAIST 전기및전자공학부 임시현

1. 문제 인식 (Problem) : 왜 CV 기반 택시 픽업 합류 가이던스 Agent가 필요한가?

문제 인식(Problem): “배차”가 아니라 “합류(픽업)”가 병목이다

택시 호출은 이미 거대한 트래픽이지만, 기사–승객이 ‘만나는 순간’이 흔들리면 호출·배차가 잘 돼도 취소·지연·민원·불법정차·혼잡으로 비용이 발생한다. 특히 공항/심야/번화가처럼 규칙과 동선이 복잡한 곳에서 병목이 극대화된다.

- 시장 규모(도시 단위 트래픽): 서울 기준 택시 일평균 747,879건, 건당 평균 13,221원 수준(서울시 자료)으로 택시는 도시의 일상 인프라로 기능한다.
- 수요 집중 구간의 실패: 카카오 T 택시는 평균적으로 탑승 성공률 94%를 공개했으며(= 실패가 존재함을 의미함), 피크 수요일에는 연간 최대 호출일 성공률이 71.3%~83.4% 범위로 공개되어 특정 상황에서는 10번 중 2~3번이 실패할 수 있음을 시사한다.
- 공항은 더 ‘어려운 픽업 환경’: 인천공항 국제선 여객은 2024년 7,066만 명으로 역대 최대를 기록하여 도착→지상교통 전환 구간의 혼잡이 구조적으로 커진 상황이다.

손실은 단순 불편이 아니라 거래 손실이다. 서울 통계만 기준으로도 “탑승 실패”가 6%p 개선되면 약 45,000건/일이 추가로 성사되며, 이는 약 1.94억 원/일(≈216억 원/년)의 거래 스케일에 해당한다.

왜 택시를 놓치는가? (핵심 원인)

핀은 정확하지만 여전히 실패가 발생한다. 문제는 좌표가 아니라 현장 맥락에 있다.

1. 장소 규칙/동선 복잡성: 공항·대형 시설은 게이트/베이/차로/정차 가능 영역이 분절되어 “정차는 되었지만 승객이 못 찾는” 상황이 자주 발생한다.
2. 서로가 보는 장면의 차이: 승객은 “출구 간판 앞”, 기사는 “우측 1차로 연석”을 기준으로 움직여 소통이 어긋난다.
3. 커뮤니케이션 비용: 소음/혼잡/언어 장벽(특히 외국인 도착객) 환경에서는 전화·채팅을 반복하게 되어 실패 확률이 높아진다.
4. 자율주행 시대의 새로운 공백: 로보택시는 ‘기사의 임기응변’ 여지가 줄어들고, 사용자는 픽업 위치를 더 자주 혼동하게 되는 문제가 제기된다.

Why Now?

- 호출 시장 성장과 함께 공항의 라이드헤일링/연석 혼잡 이슈는 구조적으로 커지는 방향으로 이동하고 있다(Rutgers, 2019).
- 이제 대부분의 자동차는 엣지 컴퓨팅 디바이스로 기능한다. 운전자의 스마트폰 또는 자체 자율주행 컴퓨팅 모델을 장착한다.
- 자율주행 확산 국면에서는 ‘주행’만큼 ‘합류 UX’가 서비스 품질을 결정하는 구간이 된다.

우리 Agent가 해결하는 것

- **택시가 정차하면 대시보드 캠이 사진을 촬영하고 CV 및 Language 모델을 통해 현장의 맥락을 판단한다.**
  - 모든 지시는 택시가 처한 맥락(GPS, 대시보드 캠)을 기준으로 작동하며, 사용자는 Agent의 지시를 따르기만 한다.
- “핀 위치” 대신 현장 앵커 기반 지시를 제공한다.

  예) “5번 출구로 나오신 뒤, 티웨이 항공 표지판 앞으로 이동한다. 차량은 우측 1차로 연석에 정차해 있다.”
- 혼잡/정차불가 상황에서는 대체 합류 지점을 자동으로 제안한다(“30m 뒤 5C 베이로 이동한다.”).
- 외국인 사용자를 위해 언어를 자동 전환하고, 짧고 단호한 문장으로 안내한다.
- GPS 정보 + 스트릿뷰 기능과 연동하여 AR의 형태로 택시의 위치를 보여준다.
- 최종 제품은 유/무인 택시에서 즉시 동작 가능한 픽업 안내 레이어(Pickup Instruction Layer)로 설계하며, 택시 호출 과정에서 발생하는 “현장 합류”를 안정화하는 표준 모듈로 제공한다. 자율주행 차량 장착은 중장기 목표로 두되, 초기 시장에서는 유인 택시를 통해 수익화와 데이터 학습을 우선 확보하도록 구성한다.

UX map

단계	승객(앱)	기사/차량	백엔드
1. 호출/배차	호출한다.	배차를 수락한다.	장소 규칙(공항 SOP 등), GPS를 로드한다.
2. 도착 임박	“곧 도착” 알림을 수신한다.	저속/정차 이벤트가 발생한다.	택시 Dashboard 캠이 정차 전후 N초 클립을 확보한다.
3. 정차(핵심)	“어디로 가야 하는지”를 고민한다.	실제 정차를 수행한다.	대시보드 클립을 CV 모델과 Language Model이 분석하며 앵커를 인식하고 1문장 지시를 생성한다.
4. 합류	지시에 따라 도보로 이동한다.	승객을 확인한다.	필요 시 추가 힌트를 제공한다.

단계	승객(앱)	기사/차량	백엔드
5. 탑승	탑승을 완료했음을 확인한다.	출발한다.	성공/실패 로그를 기반으로 RL 미세조정을 통해 시스템을 개선한다.

Ultimate Agent Service Flowchart (Overview, 이해에 가장 도움이 됩니다!)

구분	세부 항목
입력	- 호출/매칭 이벤트(ride_id, 언어)를 입력으로 받는다. - 차량 상태(저속/정차 이벤트)를 수집한다. - 카메라 스트림(짧은 클립): 정차 전후 N초(예: 3~5초)를 확보한다. - 기사 예외 입력(정차불가/혼잡 등)을 반영한다. - (선택) 장소 규칙(SOP) 데이터를 함께 활용한다. - 택시 GPS
계산	(1) Video Perception Stack (YOLO 등)을 활용한다. - 표지/출구/기둥/게이트/항공사 간판 등 멀티 앵커를 탐지하고 추적한다. - 세그멘테이션으로 연석/차로 경계·정차 영역을 안정적으로 인식한다. - 영상 품질(야간/역광/가림/흔들림)을 평가한 뒤 필요 시 재캡처 또는 대체 앵커를 자동 선택한다. (2) Multi-Modal Scene Understanding (무거운 VLM/Video-VLM)을 적용한다. - OCR 결과가 애매한 경우 표지가 무엇인지, 어떤 출구인지 시각·언어적으로 보완한다. - 혼잡/위험 상황을 설명 가능한 형태로 요약한다. (3) Instruction Policy Engine (학습된 의사결정)을 사용한다. - 후보 안내를 다수 생성한 뒤 성공확률/안전/ETA(추정)/불확실성 기준으로 LLM RL 미세조정을 통해 랭킹한다. - 최종 출력 형식은 1문장 + 앵커 + 이미지(GPS + 로드뷰 + AR)로 고정하여 UX를 만든다. (4) Uncertainty-aware Safety를 적용한다. - 불확실성이 높을 경우 보수적 안내(대기/재캡처/기사 이동)로 자동 폴백한다. - 정책 위반 가능성이 있는 문구는 제약 기반(Constrained) 생성으로 차단한다.
출력	- 승객 UI에는 1문장 행동 지시 + 앵커(텍스트) + 캡처(또는 짧은 GIF/스냅샷)를 제공한다. - 기사 UI에는 전송된 내용과 예외 버튼, (선택적으로) 추천 정차 포인트를 제공한다. - 서버에는 로그/리플레이, 실패모드 자동 태깅, 정책/모델 개선 큐를 축적한다.

2. 실현 가능성 (Solution) : 창업 아이템의 개발 계획

2-1) Pre-MVP Agent 학습 Loop

이 문제는 Neural Network을 이용한 전형적인 end-to-end 학습 문제로 다룰 수 없다. 따라서 Agentic한 플로우를 구축해서 현실 문제를 잘 해결할 수 있도록 했다. 또한, 위에서 언급한 MVP의 엣지 컴퓨팅을 본격적으로 시작하기 전에, Instruction 생성 Agent 플로우의 성능 및 구현 가능성을 고민해보았다.

데이터 수집 및 전처리

- Kaggle `bdd100k` 및 여러 4K/8K 유튜브 주행 영상을 수집한다.
- Laplacian 분산 기반 선명도 필터로 고품질 프레임만 자동 추출하고, 학습용 해상도와 포맷으로 일괄 전처리한다.

자동 라벨링 및 객체 탐지 모델

- **Florence-2**(Vision-Language 모델)로 차량, 보행자, 신호등, 표지판 등 객체를 탐지한다.
- **EasyOCR + Tesseract**로 platform_sign, traffic_sign 내 숫자 정보를 인식한다.
- 이를 바탕으로 `platform_sign / traffic_sign / crosswalk / traffic_light / vehicle / pedestrian` 6개 클래스를 자동 라벨링하고,

  YOLOv8 (PyTorch 기반) 으로 전이학습하여 경량 객체 탐지 모델을 학습한다.

OCR 기반 표지판 세부 정보 인식

- YOLO가 감지한 표지판 영역을 crop한 뒤, CLAHE·adaptive threshold 등 다양한 전처리와 Tesseract digits-only 모드를 조합해 플랫폼 번호·속도 제한 숫자를 안정적으로 추출한다.

Instruction 생성 로직

- 감지된 플랫폼 번호·교통 표지 숫자, 횡단보도·신호등·보행자 존재 여부를 조합해 "플랫폼 X 표지 방향으로 이동하세요"와 같은 안내 문장 후보들을 생성한다.
- 향후에는 GPS 정보(승객-차량 거리·방향)를 추가해 "어느 방향으로 몇 m 이동"까지 포함하는 정량적 안내로 확장할 예정이다.

자세한 내용은 별도의 Demo 파일에 잘 정리되어있다.

2-2) PoC & MVP

PoC를 위한 MVP는 다음과 같은 단계와 Tech stack으로 작동하도록 설계한다. 이 단계는 정확도 경쟁이 아니라 합류 KPI 개선을 유인 택시에서 빠르게 증명하고, 그 과정에서 데이터 로그를 축적하는 것을 목표로 한다.

단계	Shadow(Week 2~3)	Guided(Week 6~8)	A/B Test(Week 6~8)
입력	호출/매칭 + 정차/저속 트리거 + 스냅샷(1~3장)을 수집한다.	Shadow와 동일한 입력에 더해 승객 링크/웹뷰를 전송한다.	Control(기존 GPS+채팅)과 Treatment(Guided)를 나누어 입력을 수집한다.
계산	ROI 크롭(룰/경량 탐지) + OCR + Anchor 정규화를 수행한다.	Shadow와 동일한 계산을 수행한 뒤 템플릿 기반 안내 1문장을 생성한다.	KPI(합류시간/취소 수/못 찾는 수/문의 수)를 비교한다.
출력	로그만 저장하며 승객 안내는 제공하지 않는다.	승객 Webview에 1문장 + Anchor + 이미지 + 피드백 버튼을 제공한다.	효과 수치를 포함한 결과 리포트를 생성한다.

레이어	구성요소	기술/도구
Edge(차량)	앱/런타임	Android (Kotlin), CameraX를 사용한다.
	추론/가속	TFLite + NNAPI를 사용한다.
	ROI 처리	룰 기반 크롭 또는 경량 Detector(표지/OCR 영역만)를 사용한다.
	OCR	ML Kit Text Recognition(온디바이스)을 활용한다.
	트리거	정차/저속 이벤트 기반으로 캡처(스냅샷 1~3장)를 수행한다.
Passenger UI	전달 방식	모바일 웹(Webview/링크)로 안내를 제공한다.
	프론트	React 또는 Next.js로 구현한다.
Backend(데이터)	API 서버	FastAPI를 사용한다.
	실시간(선택)	SSE(구현이 쉬운 방식)를 기본으로 사용하고, 필요 시 WebSocket을 고려한다.
	DB/스토리지	PostgreSQL + S3(Object Storage)를 사용한다.
Analytics(초기)	분석/리포트	Metabase 또는 간단 Admin + CSV Export를 사용한다.
LLM	사용 여부	원칙적으로 사용하지 않으며, 필요 시 고정 번역 템플릿만 활용한다.

2-3) 기능·성능 차별성 및 경쟁력 확보 전략

1) 합류 실패를 Micro-funnel KPI로 "분해"하는 계측 체계

- 기존 서비스는 '배차→승차 실패'를 단일 지표로 처리해 실패 원인이 블랙박스로 남는 경향이 있다. 본 시스템은 배차 이후 0~3분 합류 구간을 단계별 KPI로 분해하여 "어디에서 왜 막히는지"를 정량화한다.
- 이를 통해 실패를 CS 라벨이 아닌 행동 가능한 병목(Intervention point)으로 전환하고, 개선 효과를 단계별로 증명한다.

2) 한국/공항/도심 로컬 환경에 최적화된 "로컬라이제이션 엔진"

- 합류 문제는 GPS 정확도보다 SOP·단속·동선·표지 체계·언어/외국인 UX와 같은 로컬 변수에 의해 더 크게 좌우된다고 본다. 따라서 국가/시설마다 로컬라이제이션을 필수 전제로 둔다.
- 본 시스템은 "글로벌 원툴"이 아니라, 장소 규칙과 시각 앵커(출구/표지/기둥/게이트)를 기반으로 안내하여 초행, 외국인, 혼잡 환경에서도 통하는 합류 방식을 제공한다.

3) '모두 원점'인 영역을 선점하는 운영형 제품: A/B·로그 내장 개선 루프

- 합류 구간은 표준 KPI/벤치마크가 거의 없는 영역이므로, 현재는 누가 먼저 측정 체계를 만들고 적절히 잘게 쪼개며 운영 루프를 굴리느냐가 승부처라고 본다. 즉, 소형 팀으로 빠르게 운영할 수 있는 우리 팀이 강점을 가진다.
- 본 시스템은 도입 즉시 실험이 가능하도록 이벤트 로그, 실패 태깅, 리포트를 내장하고, 주 단위로 가설→A/B→개선을 반복하여 한국형 표준 KPI를 선점하는 것을 목표로 한다.
- 단순한 A/B 테스트 뿐만 아니라 KPI를 Reward Function으로 갖는 LLM Instruction Post Training을 통해 사용자에게 더 적합한 Struction을 생성할 수 있다.

4) 기술적 장점: Edge-first 이벤트 추론 + 설명 가능한 안내 + 안전 Fallback

- 본 시스템은 "항상 영상 추론" 방식이 아니라, 이벤트 기반(저속/정차/합류 구간만) Edge 추론을 기본으로 설계하여 비용·발열·지연을 최소화하면서도 현장 대응성을 확보한다.
- 안내는 단순 좌표가 아니라 표지/출구/기둥 등 앵커 기반으로 제공하여 사용자가 이해 가능한 형태로 "왜 해당 위치로 이동해야 하는지"를 설명 가능하게 만든다.
- 또한 신뢰도가 낮거나 규정 위반 위험이 감지되면, 안전 정책(금지 지시 차단)과 대체 앵커/안전 모드로 즉시 전환(Fallback)하여 사고·분쟁 리스크를 줄인다.

3. 성장전략 (Scale-up): 사업화 추진 전략

3-1) 시장 진입 전략: 공항(Beachhead) → 도심 → 자율주행(미래)

핵심 전략은 "가장 돈이 많고, 실패 비용이 큰 구간(공항 픽업)"부터 정복하는 것이다.

왜 공항이 Beachhead인지에 대해 다음과 같이 정의한다.

- 공항은 "도착 직후(Arrival→Pickup)"에 언어/동선/표지판/정차 규정이 동시에 얽혀 '만남 실패(취소/대기/불법정차/CS)'가 폭발하는 구간으로 기능한다.
- 특히 인천공항은 2024년에 국제선 이용객 70.67M을 기록(+26.7% YoY)하며 2019년 기록을 상회하였고, 이로 인해 "픽업 병목"이 트래픽 측면에서 이미 임계치에 도달한 상태라고 판단한다.
- 공항 운영 주체도 이 문제를 운영 레벨 이슈로 인식하고 있으며, 인천공항은 택시 통합배차(파일럿)와 같은 제도 실험을 실제로 시작한 상태이다 (2025-11-25 기준).

  → 따라서 현재 공항은 "해결 의지 + 예산 + KPI(혼잡/정차/민원)"가 동시에 존재하는 시장이라고 볼 수 있다.

확장 경로(공항→도심→자율주행)는 다음과 같이 설정한다.

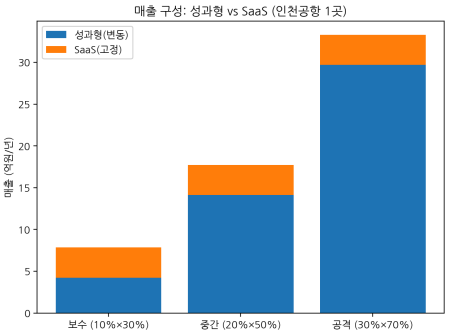
1. 공항: 표지판/게이트/기둥/차로/정차구역 등 시각 앵커가 풍부하여 CV 가이던스 효율이 가장 높게 나타나는 환경으로 본다.
2. 도심(대형 복합시설/공연장/역세권): 공항에서 검증한 "픽업 합류 가이던스"를 혼잡 핫스팟으로 확장한다.
3. 자율주행/로보택시: 차량의 주행은 블랙박스에 가깝더라도, 이용자에게 필요한 것은 '어디서 어떻게 탑승할지'라는 HMI(휴먼 인터페이스) 계층이라고 본다. 본 제품은 "자율주행 컴퓨팅 ↔ 인간" 사이의 표준 픽업 레이어로 성장하는 것을 장기 목표로 둔다.

3-2) 예상 수익 및 순현금흐름

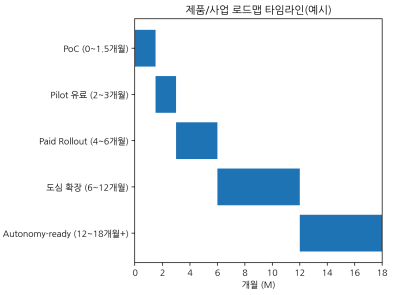
계산과정

- 연간 공항 국제여객 P = 70,670,000 (인천공항 2024 국제여객 기준)으로 둔다.
- 택시/호출 이용 비중 T ∈ {10%, 20%, 30%}로 가정한다.
- 가이던스 적용률 A ∈ {30%, 50%, 70%}로 가정한다.
- 가이던스 적용 완료콜(연)은 P×T×A로 정의한다.
- 성과형(변동) 매출(연)은 완료콜 × ₩200으로 계산한다.
- SaaS(고정) 매출(연)은 ₩30,000,000/월 × 12 = ₩360,000,000 (3.60억원)으로 설정한다.
  - 플랫폼(예: 모빌리티 앱) 관점에서 추가로 완료된 운행 1건이 만드는 기여이익은 다음과 같이 둔다.

        - 평균 운임 F = 35,000원(가정)
        - 플랫폼 수수료율 r = 10%(가정)
        - 1건당 플랫폼 매출 = F×r=3,500원
    - 연 SaaS(월 3,000만원)의 손익분기 추가 완료콜 수는:
        - 연 SaaS = 30,000,000 × 12 = **360,000,000원**
        - 필요 추가 완료콜 N = 360,000,000 / 3,500
          = **102,857건/년 ≈ 282건/일**
        - 그리고 중간 시나리오에서 가이던스 적용 완료콜이 **7,067,000건/년**이므로,
          필요한 개선률(추가 완료콜 비율)은: 102,857 / 7,067,000 = **0.01456… ≈ 1.46%**
    - 가이던스가 **완료율을 1.5%p만** 올려도(또는 취소를 그만큼만 줄여도) **연 3.6억 SaaS는 상쇄**된다.
    - 구매 저항이 너무 클 경우에는 SaaS를 아예 무료로 만들고 성과형만으로 도입 장벽을 낮춰도 된다.
- 최종적으로 총매출(연)은 성과형 매출과 SaaS 매출의 합으로 정의한다.



택시/호출 비중(T)과 가이던스 적용률(A) 조합에 따른 총매출(연)을 비교한다. 본 그래프는 3개 대표 시나리오(보수/중간/공격)에서의 연 매출을 비교하는 것을 목표로 한다. 총매출은 SaaS 고정매출(연 3.60억원)에 더해, 가이던스 적용 완료콜(P×T×A)에 비례하는 성과형 매출(완료콜×₩200)로 구성된다. 동일한 공항(인천) 환경에서도 T(택시 비중)와 A(적용률)가 증가할수록 성과형 매출이 선형적으로 커지며 총매출이 크게 확대된다고 해석한다.



단계별 매출	기간	적용률 A	완료콜(건) = (P×T)×A×(m/12)	성과형 매출(₩)	SaaS 매출(₩)	총매출(₩)
PoC	1.5개월	0%	0	0	0	0
Pilot 유료	1.5개월	5%	88,337.5	17,667,500	15,000,000	32,667,500
Paid Rollout	3개월	20%	706,700	141,340,000	90,000,000	231,340,000
도심 확장 *(공항 매출만 반영)*	6개월	35%	2,473,450	494,690,000	300,000,000	794,690,000
Autonomy-ready *(공항 매출만 반영)*	6개월	60%	4,240,200	848,040,000	420,000,000	1,268,040,000

단계별 투자금	스마트폰(일회)	데이터요금	클라우드/서버	데이터 라벨링/QA	현장 운영/출장/기타	단계 투자금 합계
PoC (3대)	3×45만 = 1.35M	3×2만×2 = 0.12M	0.5M	1.0M	0.5M	3.47M
Pilot (20대)	20×45만 = 9.0M	20×2만×2 = 0.8M	1.0M	2.0M	1.0M	13.8M

단계별 투자금	스마트폰(일회)	데이터요금	클라우드/서버	데이터 라벨링/QA	현장 운영/출장/기타	단계 투자금 합계
Rollout (100대)	추가 80대 = 36M	100×2만×3 = 6M	3M	6M	8M	59M
도심 확장 (300대)	추가 200대 = 90M	300×2만×6 = 36M	12M	20M	74M *(운영/외주 포함)*	232M
Autonomy (500대)	추가 200대 = 90M	500×2만×6 = 60M	24M	30M	158M *(운영/외주 포함)*	362M

단계	총매출(₩)	투자금(₩)	순현금흐름(₩)
PoC	0	3,470,000	-3,470,000
Pilot	32,667,500	13,800,000	+18,867,500
Rollout	231,340,000	59,000,000	+172,340,000
도심 확장 *(공항 매출만)*	794,690,000	232,000,000	+562,690,000
Autonomy-ready *(공항 매출만)*	1,268,040,000	362,000,000	+906,040,000

- 클라우드/서버 비용은 AWS 가격표(Lightsail 월 과금, CloudWatch 로그 /GB, S3 GB당 과금)와 모니터링(Sentry) 비용을 합산하여 상한(cap)을 설정한다.
- 데이터 라벨링/QA 비용은 MTurk·Ground Truth의 per-task 구조와 2026년 최저임금 시급을 인건비 하한으로 두고, 작업 속도와 QA 비율을 반영하여 산정한다.
- 현장 운영/출장/기타 비용은 2026년 최저임금(현장지원 하한)과 국내 SW 인력 일 단가 공개자료를 기반으로 인·월 투입량을 곱해 산정한다.

4. 팀 구성 (Team): 대표자 및 팀원 구성 계획

4-1) 대표자 보유 역량(경영·기술·노하우·네트워크)

본 과제는 “모델 성능 경쟁”이 아니라, 현장(공항/도심)에서 바로 동작하는 픽업 가이던스 시스템을 빠르게 만들고 A/B 테스트로 KPI 개선을 증명한 뒤 도입까지 연결하는 문제라고 정의한다. 따라서 대표자(공동창업자)는 연구 중심 역량보다 제품화·실험·도입 중심 역량을 핵심으로 둔다.

- 임시현 (Product/Business Owner)
  - KAIST 전기및전자공학부 + 물리학부 복수전공
  - KAIST AI 학술동아리 Include 소속
  - AI Agent 개발 경험 보유
  - KAIST 기술경영학부 형용준 교수님과 별도의 스타트업 진행중
  - 문제를 정의하고 KPI(합류시간/취소율/CS/불법정차 등)를 설계하며, PoC 운영과 파트너 커뮤니케이션(플랫폼/공항/택시 네트워크)을 담당한다.
  - “실험 가능한 제품”으로 빠르게 범위를 좁히고(Shadow→Guided→A/B), 개선 효과를 계약 근거로 전환하는 역할을 수행한다.
- 김지혁 (Tech/Edge/CV Owner)
  - KAIST 기술경영학부 + 전산학부
  - KAIST AI 학술동아리 Include 소속
  - KAIST 개발그룹 SPARCS 소속
  - KAIST 창업동아리 KE 소속
  - AI Agent 및 AI 서비스 개발 경험 다수
  - 엣지 기반 인식 파이프라인(Detect→OCR→Anchor→1문장 안내)을 구현하고 지연·비용·프라이버시 측면에서 최적화한다.
  - 실패모드/불확실성 기반 안전 폴백(보수적 안내, 위험 문구 차단)과 로그 기반 개선 루프를 구축한다.
- Advisor: KAIST 전산학부 홍승훈 교수님
  - 홍승훈 교수님께서는 Vision and Learning Laboratory의 Director이시며, 최소 감독을 통해 Vision 모듈이 최대한 많은 Context와 개념을 이해하는 것을 연구하신다. 즉, 우리의 문제에 대한 많은 학계의 인사이트를 가지고 계시며, 우리가 해결하려는 문제가 실생활적인 가치 뿐만 아니라 일반 과학 문제로 추상화 하는 과정을 도와주신다.

4-2) 보유/예정 장비·시설 및 직원 역량(확보 계획 포함)

소규모 팀의 인력·컴퓨팅 제약을 약점이 아니라, 현장형 설계 원칙(신뢰성/지연/비용/프라이버시)을 강화하는 방향으로 전환한다. 초기 PoC/MVP는 “거대 학습”이 아니라 스마트폰 기반 엣지 추론과 최소 서버, 로그/리플레이로 구성한다.

- 초기 보유/예정 장비(최소 구성, 파일럿에 최적화함)

- Android 스마트폰 2~4대(차량 거치 + 상시 전원)를 준비하여 스냅샷 캡처, 온디바이스 OCR, 경량 탐지를 수행한다.
- (선택) Jetson Orin/Coral 1~2대를 확보하여 야간·역광 등 하드 케이스에서 성능을 검증하고 최적화를 진행한다.
- 서버(경량)는 FastAPI/NestJS + PostgreSQL + S3(Object Storage)와 대시보드(Metabase 등)로 구성한다.

- 역량 확보(채용/협업 우선순위)는 다음과 같이 설정한다. 연구 인력보다 배포/운영/실험 역량을 우선 확보하는 것을 원칙으로 한다.
  1. Mobile/Client: 승객 Webview/링크 전달, 기사 UI, 현장 UX 구현을 담당한다.
  2. Edge/CV: 온디바이스 최적화(TFLite/NNAPI)와 안정성/배터리/발열 관리를 담당한다.
  3. Ops/BizDev: 현장 SOP 정리, 파일럿 운영, 파트너 커뮤니케이션을 담당한다.

4-3) 협력(예정) 파트너/기관 및 주요 협업 내용

본 과제는 "기술 단독"으로는 성과를 내기 어렵고, 트래픽·현장 운영·정책 환경이 결합되어야 한다고 본다. 소규모 팀은 이를 파트너 레버리지(데이터·실험 슬롯·도입 채널)를 통해 보완한다.

- 모빌리티 플랫폼(Primary, 예: 카카오T 연계)
  - 파트너는 호출/매칭 이벤트, KPI 측정 환경, 실험 슬롯(A/B)을 제공한다.
  - 본 팀은 픽업 가이던스 모듈(1문장+앵커+캡처), 실패모드 로깅/리플레이, 안전 정책 레이어를 제공한다.
- 공항/터미널 운영 주체(Secondary)
  - 파트너는 픽업존 룰·사이니지·SOP, 운영 정책 및 현장 협조를 제공한다.
  - 본 팀은 curbside 혼잡/불법정차/민원 감소를 위한 운영 대시보드와 표준 프로토콜(Anchor DB, 금지 행동 정책)을 제공한다.
- 택시 네트워크(법인/조합/기사)
  - 일부 지자체가 카카오T 독점을 완화하기 위해 자체 택시 서비스를 론칭하는 흐름을 고려한다.
  - 파트너는 차량 장착, 현장 피드백, 예외 상황 데이터(정차불가/혼잡 등)를 제공한다.
  - 본 팀은 합류시간 단축을 통해 회전율을 높이고, 기사 스트레스를 줄이며, 분쟁/문의 발생을 줄이는 가치를 제공한다.

5. 카카오와 시너지

1) Kakao gets: 카카오의 코어 인프라를 강화하는 "Pickup Intelligence"

- 카카오T 전반의 합류 성공률을 개선하는 프리미티브를 확보한다.
  - 공항뿐 아니라 호텔/행사장/역/대형 단지 등 GPS만으로는 대응이 어려운 픽업존에서 픽업 성공 여부가 곧 제품 경쟁력이 된다고 본다.
  - 위의 3번 항목에서도 말했듯이, 탑승실패를 1~2%만 줄여도 우리 서비스를 SaaS로서 구매한 것 이상의 가치를 만들 수 있다.
- 카카오맵 고정밀 curbside/POI 레이어 자산을 축적한다.
  - 정차 가능 구역, 레인 규칙, 표지판/기둥 앵커 등 기존 지도에 없던 운영 데이터를 축적하여 지도 품질과 추천 능력을 직접적으로 끌어올린다.
- k.ride(외국인) 첫 탑승 성공 패키지를 통해 LTV를 상승시킨다.
  - 다국어 지원, 불안 해소, 합류 안정화는 외국인 온보딩 비용(CS/취소)을 줄이고 첫 탑승 전환율을 올리는 방향으로 작용한다고 본다.
  - k.ride는 누적 600,000+ 택시 탑승(2025.3 기준), 이용자 ~100개국이므로 성장 가능성이 크다.

2) We get: PoC를 "데이터/실험/배포"로 이어주는 실행 레버리지

- 실험 트래픽과 KPI 측정 환경(완료콜/취소/CS)을 확보한다.
- 공항·플랫폼과의 현장 운영 파트너십 진입 장벽을 완화한다. 정책/안전/리스크 대응 프레임을 카카오와 함께 설계하여 도입 속도를 높이는 구조를 목표로 한다.

3) Together: '유인 택시 기능'이 아니라 Autonomous-ready Mobility UX Layer 및 더 넓은 분야로 확장

- 자율주행이 확산될수록 "차가 왜 그렇게 움직이는지, 어디서 만나야 하는지"에 대한 설명과 안전 동선 유도가 핵심 UX가 된다고 본다.
  - if kakao 25의 자율주행 AI 실차 적용기 프레젠테이션을 보면, 카카오가 자율주행 시스템에도 많은 투자를 하고 있는 것으로 보이며, 특히 언급된 관제시스템의 Pick-up 모듈을 개선하는데에 시너지가 크다고 예상된다.
- 본 팀은 Scene-aware guidance, Trust messaging, Safety protocol을 카카오T 내부 공통 레이어로 구성하고, 공항에서 먼저 실증한 뒤 도심 및 자율주행 플릿으로 확장하는 로드맵을 설정한다.
- 또한 이렇게 사진 한장을 통해서 컨텍스트를 읽어내는 기술은 모빌리티 뿐만 아니라 다른 곳에도 확장 적용될 여지가 많다.
  - 사진 1장을 통한 배달/택배 배송 완료 검증 자동화 시스템
    - 배송기사가 찍은 사진 속 건물 숫자, 동 호수, 픽업 위치 등을 검사하여 피드백을 자동화 할 수 있다.
    - 국가물류통합센터에 따르면 한국의 물류시장은 연 5.95B개가 운송될 정도로 매우 큰데, 배송 실패의 비중을 3~5%만 잡아 연당 조단위의 손실이 나오게 된다. 즉, 매우 점유할 가치가 큰 시장이다.

- 사진 1장을 통한 산업 현장의 안전 점검 자동화 시스템
- 프랜차이즈 본점의 가맹점 검사 자동화 시스템
  - 사진 1장을 통해 진열대의 context를 읽어내고 특정 제품을 어느 레이블에 놓아야하는지 확인할 수 있다.

4) (선택) 2단계 수익/확장 옵션: 공항 도착 직후 커머스 연계

공항 도착 직후는 통신/교통/관광 등 고가치 전환이 발생하는 구간이므로, 픽업 안정화를 선행 과제로 두고 이후 단계에서 옵션 형태로 광고·커머스 인벤토리를 정교화하는 방안을 검토한다. 초기 MVP 단계에서는 전면에 두지 않는다는 원칙을 유지한다.

6. Demo

별도 문서로 데모를 첨부합니다.

Pre-MVP Demo Spec

7. Appendix

Ultimate Agent Service Tech Stack (예상)

레이어	모듈	역할	Tech Stack
차량 Edge	하드웨어	실시간 영상 인지/추론 수행	Jetson Orin / Orin Nano(현장 성능-비용 트레이드오프) + 카메라 + 기사폰(또는 차량 디스플레이)
	디바이스 OS/런타임	안정적 프로세스/재시작/로그	Ubuntu + systemd + Docker(가능하면)
	비디오 파이프라인	스트림 처리/프레임 샘플링	GStreamer + (선택) NVIDIA DeepStream
	추론 최적화	지연/비용 최소화	TensorRT + CUDA + FP16/INT8
	멀티 앵커 탐지	표지/출구/기둥/게이트/항공사 간판 등	YOLOv8/RT-DETR 계열(최적화 모델)
	세그멘테이션	연석/차로/정차 영역 분리	경량~중형 Seg 모델 + TensorRT
	추적/Temporal	비디오 기반 신뢰도 상승	ByteTrack/OC-SORT + temporal smoothing
	품질 평가/재캡처	야간/역광/가림/흔들림 대응	blur/노출 heuristic + (확장) 품질분류 모델
	온디바이스 OCR(옵션)	텍스트 즉시 추출	Transformer OCR(온디바이스) + 후처리
	Edge 보안	디바이스 인증/키관리	mTLS + 디바이스 토큰 + Secure storage
	Edge OTA	모델/앱 업데이트	Mender/Balena(선택) 또는 자체 OTA
클라이언트(Driver)	Driver App/Widget	기사 예외입력/정차불가/혼잡/추천정차포인트 확인	Android(Kotlin) + FCM 푸시 + WebSocket/SSE
	가이던스 확인 UI	“지금 이 앵커가 맞는지” 빠른 피드백	카드형 UI(앵커 텍스트/이미지/확신도/원클릭 확인)
클라이언트(Passenger)	Passenger UI	1문장 행동지시 + 앵커 + (선택) 캡처/GIF	WebView/Native + 다국어(i18n)
	지도/AR 안내	GPS+로드뷰+AR 오버레이	Map SDK(카카오/구글/애플 중 택1) + (선택) ARCore/ARKit
	메시징/알림	탑승자 실시간 안내	푸시(FCM/APNs) + 실시간 채널(WebSocket)
서버 AI	Video-VLM/VLM	애매 케이스 이해/대체 앵커 제안	GPU Inference 서버 + 요청 기반(온디맨드) 라우팅
	Instruction Policy Engine	후보 안내 생성→랭킹→최종 선택	Ranking/Policy 모델 + Feature store(선택)
	Safety/Uncertainty	불확실성 기반 폴백/금지문구 차단	Confidence calibration + 룰/제약 생성(Constrained)
	Geo/Place Resolver	“장소 규칙(SOP) + GPS” 결합	Geofencing + POI/Zone 매칭 엔진
	Localization Engine	국가/공항별 표현·관습 반영	템플릿/i18n + 문화권별 문장 스타일 가이드
Backend/서비스	API	ride/state/instruction 트래픽 처리	FastAPI 또는 NestJS
	실시간	UI 업데이트/상태 동기화	WebSocket + Redis PubSub(또는 NATS)
	이벤트 버스	비동기 파이프라인/재처리	Kafka(확장) 또는 Redis Streams(초기)
	DB/Storage	로그/미디어 저장	PostgreSQL + S3(또는 GCS)
	룰/SOP 관리	공항별 규칙 버전관리/배포	Rule DB + Admin 콘솔 + 버전태깅
	AuthN/AuthZ	사용자/디바이스/권한	OAuth2/JWT + RBAC
	Rate limit/보호	악성요청/비용 폭주 방지	API Gateway + Rate limiting

레이어	모듈	역할	Tech Stack
	개인정보/보안	PII 마스킹/보관정책	얼굴/번호판 블러(가능하면) + Retention 정책
	통합(파트너)	모빌리티 플랫폼/공항 시스템 연동	Webhook + 파트너 API 커넥터
	정산/과금(선택)	B2B 과금/리포팅	Stripe 등(또는 내부 청구) + Usage metering
MLOps	실험/모델 레지스트리	모델 버전/재현성	MLflow + 모델 Registry
	서빙	고성능 추론 서빙	Triton Inference Server(권장)
	데이터 플라이휠	hard-case 자동 수집/재학습	이벤트 로그 + hard-case 큐 + 샘플링 전략
	라벨링/검수	앵커/텍스트/존 라벨	CVAT/Label Studio + QA 워크플로우
	CI/CD	서비스/모델 배포 자동화	GitHub Actions + Docker + IaC(Terraform 선택)
Ops/Analytics	대시보드	KPI/실패모드/커버리지 모니터링	Metabase / Superset / React Admin
	로깅/리플레이	상태 전이, anchor_id, conf, 예외 입력 추적	이벤트 로그 + 리플레이 뷰어(내부 툴)
	관측성(Observability)	장애/지연/비용 모니터링	Prometheus + Grafana + Sentry
	A/B 테스트	문구/정책/모델 비교	Feature flag + 실험 프레임워크
LLM	문장 품질/설명 생성	다국어·자연화·설명 가능한 요약	LLM + 캐시/비동기 처리(큐)
	정책/프롬프트 관리	프롬프트 버전관리/가드레일	Prompt registry + 정책 템플릿
	RL/후학습	KPI 기반 정책 고도화	KPI를 Reward로 정의한 RL Post-Training

— end of file —