

# Manion 전체 시스템

## 1.전체 flowchart



[사진1-시스템 Flowdiagram]

## 2.단계별 핵심아이디어,방법,연결,데이터타입

### 1) 입력 수집

- **핵심 아이디어:** 모든 문제 입력을 텍스트/이미지 형태에서 표준화된 데이터로 변환한다.
- **방법:** 이미지 문제는 OCR(Mathpix 등), 텍스트 문제는 정규화 처리, 필요시 언어 번역을 거친다.
- **이전 연결:** 원시 데이터(학생 교재, 시험지 등) → 처리 시작점.
- **데이터타입:** PNG/JPG/PDF, TXT, JSON(raw problem).

---

### 2) 구조화(추론/계산 계획)

- **핵심 아이디어:** 문제를 ToT(추론 트리)와 PoT(계산 코드)로 해체해 구조적 표현을 만든다.
- **방법:** LLM 기반 ToT 생성, 코드 변환기 기반 PoT 생성, Section/Step 단위로 분해.
- **이전 연결:** 입력 수집 단계의 문제 텍스트를 분석하여 논리 구조화.
- **데이터타입:** JSON(ToT 노드/엣지), Python 코드(PoT), 중간 step 리스트.

---

### 3) 검증

- **핵심 아이디어:** 계산과 추론의 정확성을 CAS 기반으로 보장한다.
- **방법:** Sympy를 1차로 실행 → 실패 시 WolframAlpha 등 외부 CAS로 대체 검증.
- **이전 연결:** 구조화된 PoT와 ProofTape 초안을 입력으로 사용.
- **데이터타입:** Sympy Expr, JSON(검증 결과/에러 로그).

---

### 4) IR 생성(.mni)

- **핵심 아이디어:** 문제, 구조, 검증, 시각화를 하나의 파일에 패키징한다.
- **방법:** 문제/메타데이터/ProofTape/Visual 지시어를 JSON Schema에 맞게 저장.
- **이전 연결:** 검증된 ProofTape와 시각화 지시어를 포함해 최종 IR 생성.
- **데이터타입:** .mni (JSON 기반 IR).

---

### 5) 스케줄링/캐싱

- **핵심 아이디어:** 동일 작업을 반복하지 않고, 병렬·증분 실행을 가능하게 한다.
  - **방법:** 해시 기반 캐시 키 생성 → 작업 큐에 등록 → 캐시 히트 시 재계산 스킵.
  - **이전 연결:** .mni 파일의 hash\_key를 활용해 효율적 실행 관리.
  - **데이터타입:** Redis/Key-Value 캐시, 큐(Job JSON).
-

## 6) 렌더(시각화 백엔드)

- **핵심 아이디어:** 추론과 계산 결과를 시각적으로 전달 가능한 영상·그래픽으로 변환한다.
- **방법:** Manim으로 수학 애니메이션, GeoGebra/Three.js로 그래프·3D 도형을 렌더링.
- **이전 연결:** .mni 파일의 visual.sections 지시어를 실행.
- **데이터타입:** mp4, PNG, WebGL 객체.

---

## 7) 조립/후처리

- **핵심 아이디어:** 분리된 렌더링 결과를 하나의 완성된 학습 콘텐츠로 합친다.
- **방법:** partial 영상 조립, 자막·하이라이트·오디오 합성, 엔드카드 추가.
- **이전 연결:** 렌더링 단계에서 생성된 partial outputs을 결합.
- **데이터타입:** mp4(final), srt(자막), JSON(편집 메타).

---

## 8) 배포

- **핵심 아이디어:** 학습자/연구자가 접근 가능한 플랫폼에 자동 업로드한다.
- **방법:** S3·Cloud·Notion·전용 Hub API를 통해 배포, 메타데이터 인덱싱 포함.
- **이전 연결:** 조립된 최종 산출물을 전송.
- **데이터타입:** mp4 URL, JSON(메타데이터 인덱스).

---

## 9) 관측

- **핵심 아이디어:** 성능과 품질 지표를 추적해 파이프라인 최적화를 지원한다.
- **방법:** 로그 수집, 검증 통과율·렌더링 시간·캐시 히트율 대시보드화.
- **이전 연결:** 배포 후 사용자 데이터 및 실행 로그 분석.
- **데이터타입:** 로그(JSON/CSV), 시계열 DB(metrics).

---

## 10) 반복/수정

- **핵심 아이디어:** 자연어 프롬프트나 JSON diff로 빠른 수정과 증분 렌더링을 제공한다.
- **방법:** 입력된 diff 적용 → .mni 일부 갱신 → 변경된 섹션만 재렌더링.
- **이전 연결:** 관측 단계에서 확인된 오류·피드백을 반영.
- **데이터타입:** JSON Patch, 수정된 .mni, 증분 mp4.

# □ Manion MVP 범위 정리

## 1. 입력 수집

- MVP: PDF 수학문제 입력만 지원.
- 확장: 손글씨 이미지(OCR) 문제까지 지원.

## 2. 구조화

- MVP: 프롬프트 기반 단방향 풀이 구조화.
- 확장: PoT 기반 다중 풀이 경로 자동 생성.

## 3. 검증

- MVP: Sympy 기반 1차 검증만.
- 확장: WolframAlpha 등 외부 CAS fallback 추가.

## 4. IR 생성(.mni)

- MVP: 제한된 변환 템플릿 사용.
- 확장: 다양한 템플릿/스키마 확장.

## 5. 스케줄링/캐싱

- MVP: 생략.
- 확장: 동일 작업 캐싱, 병렬 처리, 증분 실행 지원.

## 6. 렌더

- MVP: .mni → GPT 변환 → Manim 코드 → 영상 생성.
- 확장: .mni → 직접 영상화 로직 구현 (GPT 불필요).

## 7. 조립/후처리

- MVP: 렌더 단계와 통합 수행.
- 확장: 별도 단계로 분리, 자막·하이라이트·오디오 추가.

## 8. 배포

- MVP: 웹사이트에 mp4 자동 업로드.
- 확장: Notion/Hub/API 등 멀티플랫폼 배포.

## 9. 관측

- MVP: 결과값 정량적 관측 (성공률·시간 등).
- 확장: 대시보드/실시간 모니터링.

## 10. 반복·수정

- MVP: 없음.
- 확장: 자연어 프롬프트/JSON diff 기반 증분 수정·재렌더링.

---

+확장: 사용자가 영상화를 .mni 확장자를 이용해 스스로 개발+조정할 수 있는 커뮤니티 시스템 만들기

\*MVP의 목적:

- 1)정확도, 배치, 범위, 논리 면에서 gpt를 활용하여 기존 영상생성기와 차별두기.
- 2).mni데이터를 만들어 수학영상생성화의 기초 틀 만들기.