

제6회 「공군 창의·혁신 아이디어 공모 해커톤」 기획서 [자유공모]

응모분야	작전지원
팀명	CUBIONS(Cube+Unions)
프로젝트명	짤려도 좋아! 4년짜리 업무, 한 달만에 끝내버리자

1. 아이디어 선정배경

□ 개 요

- 전시 보충소요기준 연구 업무를 자동화하고 시로 지능화한 산출 프로그램 개발을 추진

□ 전시 보충소요기준 연구 소개 및 중요성

- ‘전시 보충소요 기준’이란?

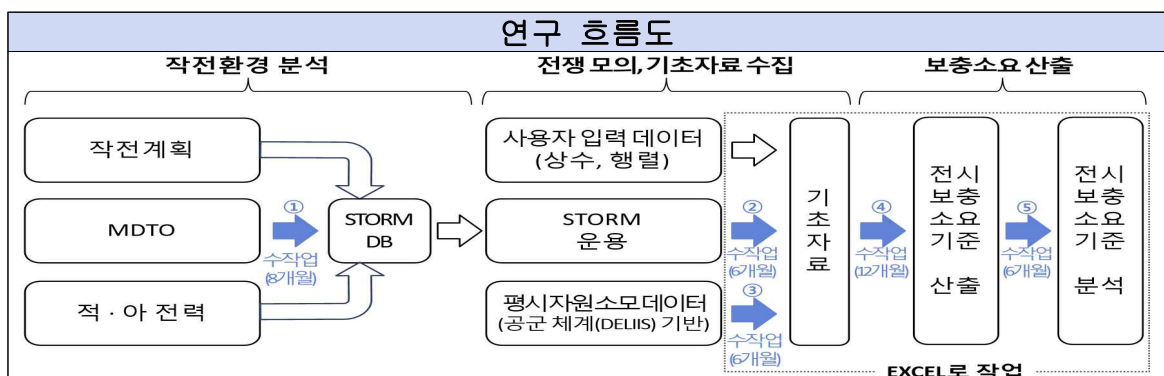
- 보유한 인력, 탄약 및 군수자원 등이 전투 피해로 인해 부족하게 될 경우를 대비하여, 작전 단계별 전투력 복원을 위해 필요한 손실 인원과 자원 등의 양을 산출한 보충소요의 기준
 - 모의분석모델(STORM)¹⁾을 활용하여 과학적이고 정량적인 기법으로 양과 계수를 산출
 - 합참 주관 하에 각 군 및 해병대에서 각 자원에 대해 2~3년 주기로 연구를 수행
- * 연구대상 자원: 총 6개 자원 (인원, 장비(항공기), 유류, 탄약, 물자, 수리부속)

- 중요성

- 평시: 훈련 및 연습 간에 탄약, 유류 등 필수 자원의 보충 소요 판단의 근거로 사용
- 전시: 비축 계획, 산업 동원계획 등 군수지원 계획의 기준 및 근거자료로 활용
- 군 자원 관련 각종 정책 및 사업 문건에 대한 기준자료 및 결심 지원

➡ 전시 작전계획을 구체화하고, 계획 시행 간에 신뢰성을 높이는 군의 중요한 업무

□ 전시 보충소요기준 연구 기존 수행절차



- ①: MDTO²⁾를 입력한 Excel 자료를 STORM DB로 수작업 입력 (8개월 소요)
- ②, ③: STORM 운용 결과 자료와 공군 평시 자원 소모 데이터를 산출 프로그램 기초자료로 변환 (1개 자원 당 2개월, 총 6개 자원 12개월 소요)
- ④: 기초자료를 활용하여 함수식 작성(Excel 사용)하여 전시 보충소요기준 산출 (1개 자원 당 2개월, 총 6개 자원 12개월 소요)
- ⑤: 전시 보충소요기준 산출 결과를 표와 그래프로 작성하고, 자원의 양과 계수의 타당성 분석 (1개 자원 당 1개월, 총 6개 자원 6개월 소요)

1. 아이디어 선정배경(계속)

□ 기존 전시 보충소요기준 연구의 문제점

- 수작업으로 Excel 입력, 산출 공식 작성 및 기준 분석 과정에 장기간 소요
- Excel 수작업은 오류 가능성이 있으며, 오류를 파악, 검수하는 과정에 장기간 소요
- 작전계획 및 작전환경(전력 등)에서 변화하는 조건을 즉각적으로 반영하기 어려움 (모델 재운용, 산출값 재계산 필요 등)

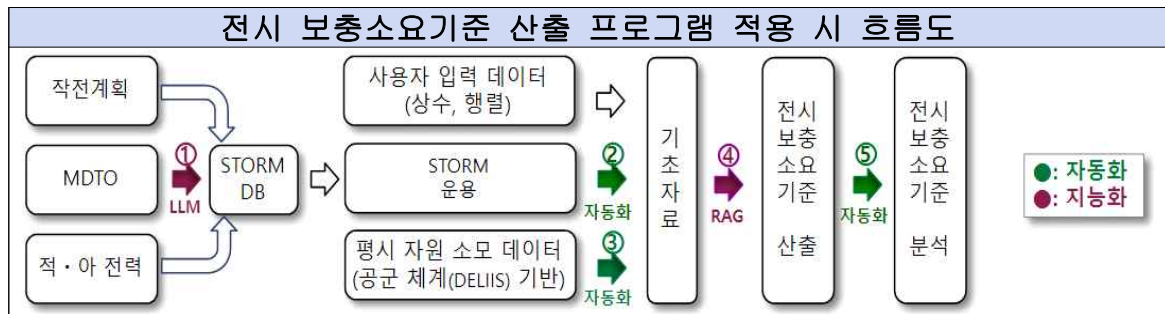
□ 발전방향

- 연구 과정을 자동화하고 프로그램을 지능화(AI)한 ‘전시 보충소요기준 산출 프로그램’의 자체 개발 추진
 - 자동화: ②, ③, ⑤ 과정을 자동화하여 업무 시간을 단축하고, 오류 가능성 최소화
 - 지능화: ① AI(LLM)를 활용해 자동화, ④ AI(RAG)로 더 정확한 공식/결과 산출
- 프로그램 개발을 통해 연구 기간 단축, 국방 예산 절감, 산출 데이터의 정확도 향상 도모

2. 아이디어의 주요 특징(핵심 기술)

□ 아이디어의 목표

- 전시 보충소요기준 산출 프로그램을 두 단계로 개발
 - 자료입력/결과생성 과정을 프로그램으로 자동화: ②, ③, ⑤ (자체개발 65% 완료)
 - AI를 접목하여 프로그램을 지능화: ①, ④ (개발을 위한 기초자료 수집 중)



□ 자동화에 필요한 도구, 핵심기술, 주요 특징 및 활용사유(②, ③, ⑤)

- 게임엔진
 - 게임 개발에 필요한 기능 및 도구의 모음으로, 화면 제작(GUI), 물리적 효과, 소리 등을 직접 기술 구현할 필요 없이 필요한 기능만 적용하여 빠른 개발이 가능
 - 활용 사유: 복잡한 코드 없이 쉽게 화면을 구성 및 디자인
- Python
 - 간결한 문법으로 빠르게 코딩할 수 있어 서버구축, 데이터(자원 등) 분석, 인공지능을 쉽게 제작할 수 있는 코딩 언어
 - 활용 사유: Python을 통해 서버를 구축하면 동일한 언어로 작성된 데이터의 분석과 인공지능 코드 사용 가능
- HTTP 통신
 - 정해진 규칙을 통해 요청과 응답을 통신(RESTful API 활용)
 - 활용 사유: 게임엔진과 서버의 연동을 위함

1) STORM(Synthetic Theater Operations Research Model, 전구급 분석모델): 합동전장 상황 하 항공우주작전 결과분석을 위한 전구급 분석모델로서 美에서 개발하여 공군(韓)은 '06년 도입

2) MDTO(Multi Domain Taking Order, 다영역임무명령서): 연합사령관의 작전목표 달성을 위해 가용한 다영역(공중, 지상, 해상, 사이버, 우주) 화력과 능력을 통합, 선정된 표적을 무기효과와 우선순위를 고려하여 타격하기 위한 명령서

2. 아이디어의 주요 특징(핵심 기술)(계속)

□ 지능화(AI 접목)에 필요한 핵심기술, 주요 특징 및 활용사유(①, ④)

○ LLM (Large Language Models)

- 사람의 언어 텍스트를 이해하고 생성하는 인공지능 (ex. Chat GPT)
- 활용 사유: STORM DB의 입력자료(MDTO)는 일일이 수작업으로 코드를 작성해야 하나 ('23년 연구 시 8개월 소요), LLM을 이용하면 몇 개의 DB코드만 직접 작성하여 전이학습³⁾을 통해 코드 작성 시간 대폭 단축 가능 (→약 0.23개월)

○ RAG (검색 증강 생성)

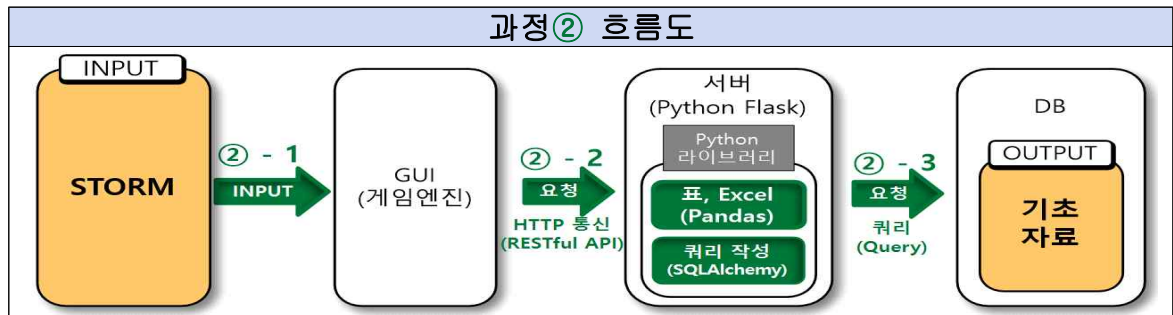
- 웹, DB, PDF 등에서 검색한 정보를 LLM에 제공하여 답변의 정확도 향상
- 활용 사유: 군 정보와 같이 LLM에 학습시킬 외부 데이터가 적은 경우, PDF 등으로 정보를 직접 제공하여 적은 데이터로도 정확한 답변 가능

3. 아이디어 구체화(현실화) 추진전략

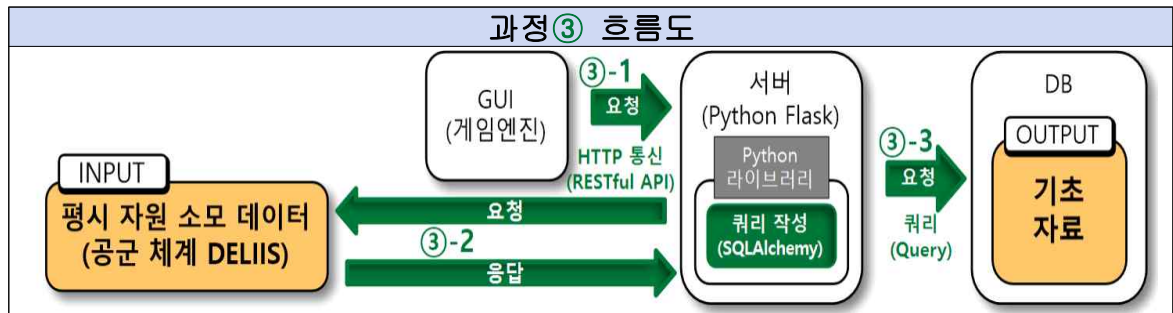
□ 자동화 단계 시나리오 : 자료를 입력하고 분석결과를 생성하는 과정을 프로그램으로 자동화

○ 운영방안

- ②: STORM 운용 결과값을 전시 보충소요기준 산출 프로그램 DB로 자동 입력



- ②-1: STORM 결과값을 화면(GUI로 구현)에 입력
- ②-2: HTTP통신으로 서버 내 Pandas를 활용하여 표로 변환
- ②-3: 변환된 표를 쿼리⁴⁾로 전달받아 기초자료로 자동 입력
- ③: 군 체계(DELIIS: 국방군수통합정보체계)를 연동하여 평시 자원 소모 데이터를 확보하고 전시 보충소요기준 산출 프로그램의 기초자료를 자동 입력

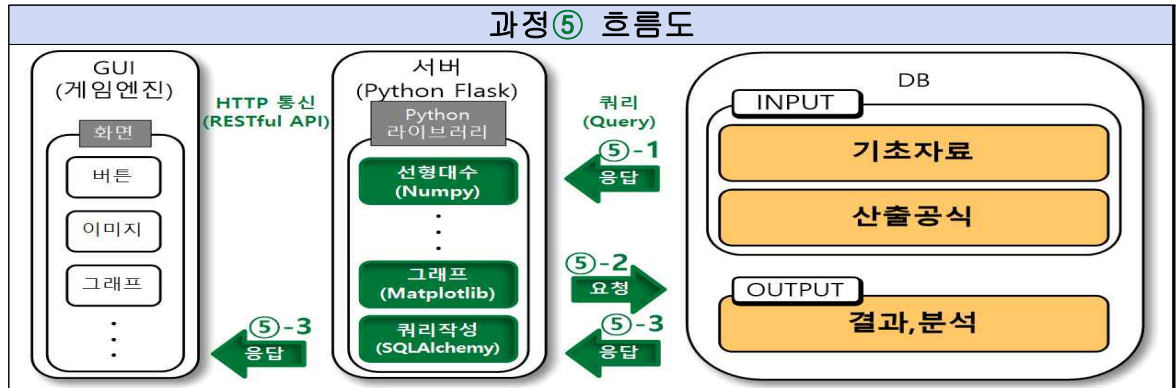


- ③-1: 화면(GUI)에서 군 체계로 데이터 요청 버튼 클릭
- ③-2: 서버에서 군 체계에 평시 자원 소모 데이터를 요청 및 응답
- ③-3: 전시 보충소요기준 산출 프로그램 기초자료에 자동 입력

3) 전이학습: 이미 학습된 AI를 활용하여 특정 부분만 추가학습을 하는 기법. 추가학습에 필요한 데이터 분량과 시간을 줄이면서도 높은 정확도를 유지함
4) 쿼리(Query): DB에서 원하는 정보를 가져오는 명령 코드

3. 아이디어 구체화(현실화) 추진전략(계속)

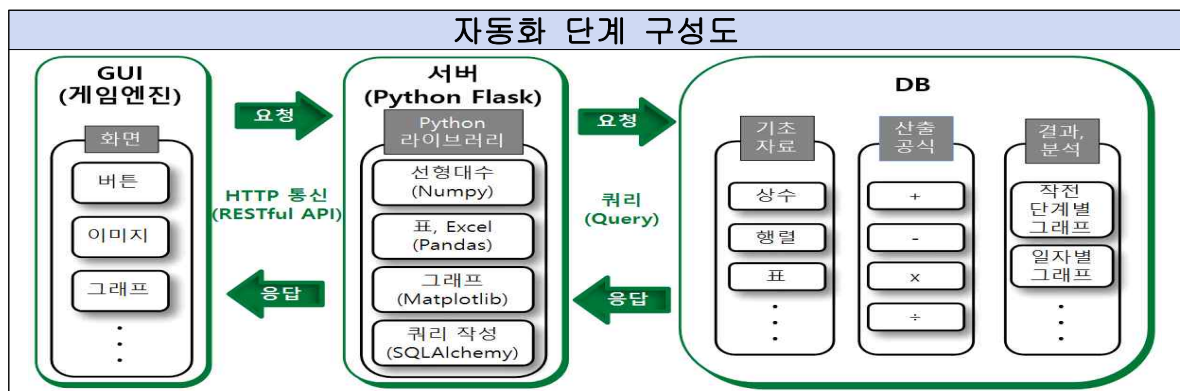
- ⑤: 자원의 양과 계수를 체계적으로 분석하기 위해 표, 그래프를 자동 생성



- ⑤-1: 기초자료 DB와 산출공식을 서버로 불러와 선형대수(Numpy)로 계산
- ⑤-2: Matplotlib(계산 및 그래프 기능 탑재)로 그래프 자동생성 및 DB에 저장
- ⑤-3: 생성된 그래프를 화면에 출력

○ 구현방안

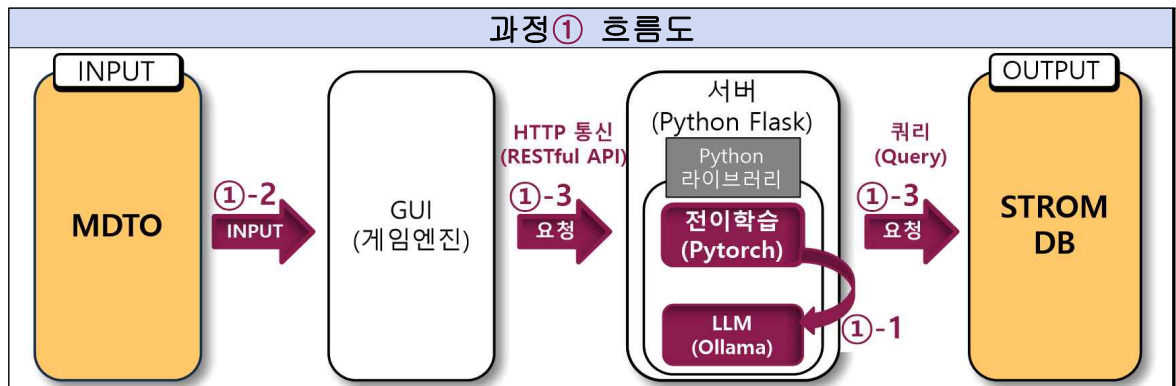
- 화면구성(GUI): 게임엔진을 활용하여 직관적으로, 사용하기 쉽게 GUI 제작
- 서버, DB 구축: Python 언어로 개발된 서버(Flask)를 사용하며 DB는 Query를 통해 조작



□ 지능화 단계 시나리오 : 단계별 자료 전환 및 결과 산출과정에 AI를 접목하여 프로그램을 지능화

○ 운영방안

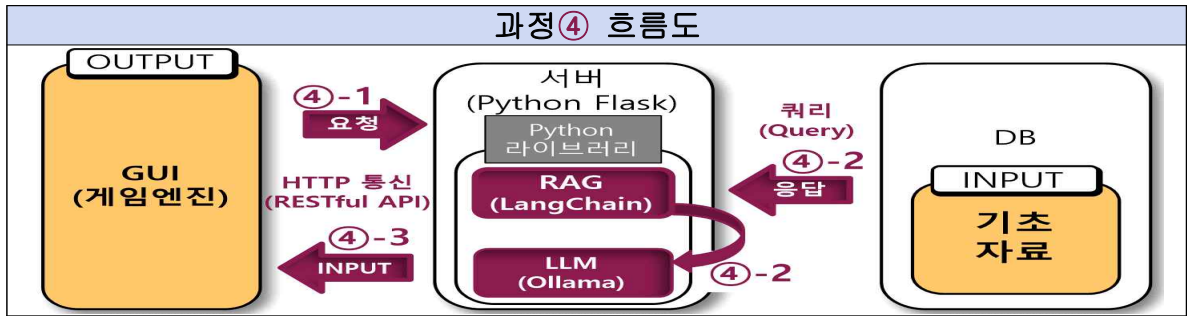
- ①: 전이학습 된 LLM을 통해 MDTO 입력데이터를 STORM DB로 변환



- ①-1: 서버의 Pytorch를 통해 과거 MDTO자료를 LLM에 전이학습 진행
- ①-2: 최신 MDTO 자료를 화면에 입력
- ①-3: LLM을 통해 STORM DB로 자동 변환

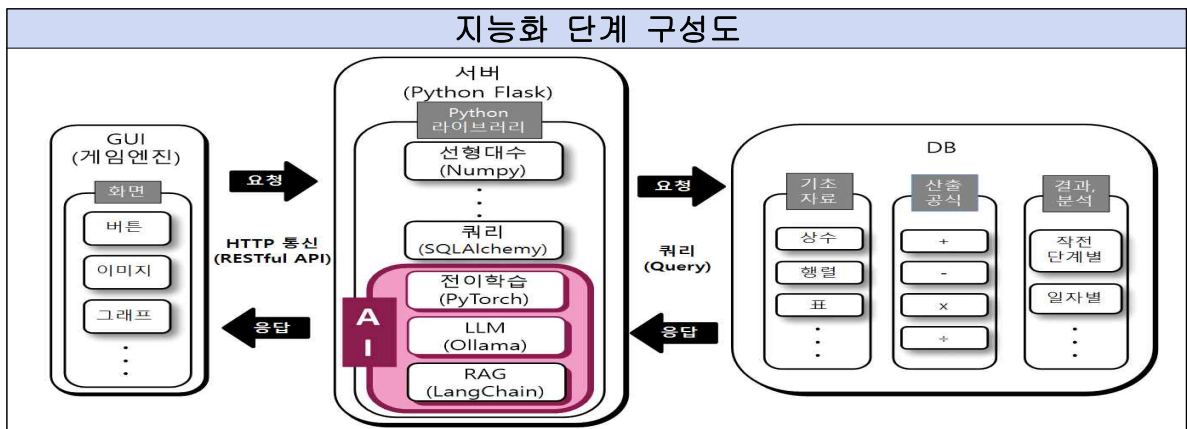
3. 아이디어 구체화(현실화) 추진전략(계속)

- ④: RAG를 통해 기존 보충소요기준 연구의 산출논리 분석 및 연구 동향 질문, 특수 조건(계절 등)을 산정공식에 어떻게 추가/적용해야 하는지 조언
 - ④-1: 기존 연구의 산출논리 및 연구 동향을 화면을 통해 질문
 - ④-2: 이전 연구자료를 제공하여 서버의 RAG로 출력될 답변의 배경지식 습득 및 LLM에 전달
 - ④-3: LLM으로 화면에 답변 생성



○ 구현방안

- AI 기술 활용: LLM과 RAG 기술로 ChatGPT와 같은 검색 및 답변 기능 확보

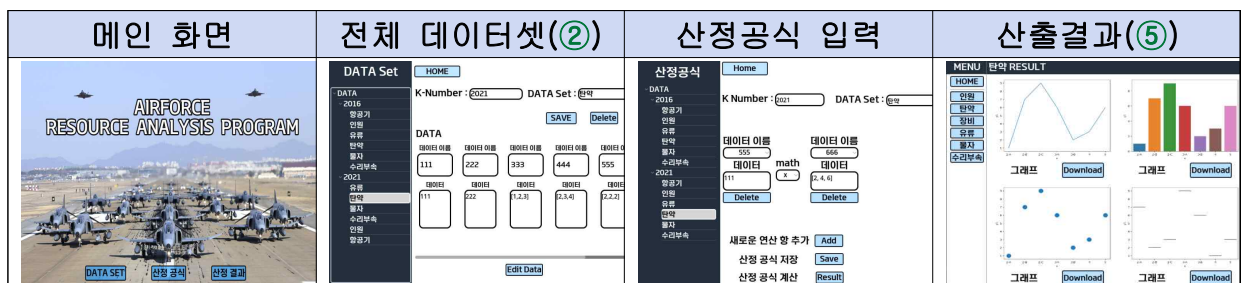


4. 아이디어 구체화(현실화)를 위한 세부방법

□ 자동화 단계

○ 필요요소 구현

- 화면구성(GUI): Godot 게임엔진을 활용하여 직관적으로, 사용하기 쉽게 GUI 제작



• 서버구축

- Python으로 작성된 Numpy(선형대수), Pandas(표, Excel), Matplotlib(그래프) 라이브러리를 사용하기 위해 Python 언어의 서버(Flask)를 사용
- Python의 Flask 서버는 GUI와 HTTP통신(RESTful API)으로 연동

- DB 연동: Python의 SQLAlchemy에서 쿼리(Query)를 사용하여 DB와 서버 연동

- 군 체계(DELIIS) 연동: 평시 자원 소모 데이터를 HTTP통신을 통해 산출 프로그램에 확보

4. 아이디어 구체화(현실화)를 위한 세부방법(계속)

□ 지능화 단계

○ 필요요소 구현

- 서버: Python 라이브러리에 Pytorch(전이학습), Ollama(LLM), LangChain(RAG) 추가
- LLM(AI) 기능 추가
 - 오픈소스(무료 배포 및 사용 가능)인 LLaMA3 모델을 이용
 - LLaMA3 모델 2개 사용 (1. STORM DB 자동생성, 2. RAG를 통한 답변)
- RAG(AI) 기능 추가
 - LangChain으로 LLaMA3(LLM) 기반에 산출 프로그램 DB 연동

□ 예측되는 장애요소 해결방안

○ LLM 전이학습

- 장애요소: LLaMA3 모델의 전이학습을 위해 GPU VRAM은 24GB이상 필요
- 해결방안: 보유 중인 Nvidia A100 GPU로(VRAM: 40GB) 전이학습 진행

○ RAG

- 장애요소: AI가 관계형 DB⁵⁾ 이해 불가
- 해결방안: AI가 이해할 수 있는 숫자 형태로 벡터 DB⁶⁾ 변환 추가

○ 비밀자료 활용에 따른 산출물(DB) 관리

- 장애요소: II급 비밀자료를 기반으로 결과가 산출되므로 보안 통제 필요
- 해결방안: 산출물(DB) 파일을 암호화하고, II급 비밀로 분류 및 별도 관리

5. 기타 아이디어 구체화를 통한 효과 등 기술

□ 아이디어를 통한 기대효과

○ 작업 기간 단축 및 예산 절감

- 총 작업 기간 단축: 기존 38개월 → 개선 후 1.23개월
 - STORM DB 입력(①): 기존 10명 작업, 8개월 소요 → 1명 작업, 0.23개월(7일) 소요
 - 기초자료 입력(②, ③, ④, ⑤): 기존 3명 작업, 30개월 소요 → 1명 작업, 1개월 소요
- 총 예산 절감액: 681,138,843원(3년 기준, 매년 227,046,281원 절감) + α
 - * α: 기존에 투입된 인원이 다른 업무를 하게 됨으로써 발생하는 추가 이익
 - STORM DB 입력(①): 321,943,743원(기존 - 개선 후)

구 분	기 존	개선 후
평균 월급(계급) × 인원	4,035,900(소령 7호봉) × 10명	4,035,900(소령 7호봉) × 1명
작업 소요기간('23년 작업 시)	8개월	약 0.23개월(7일)
합 계	322,872,000원	928,257원

- 기초자료 입력, 산출 및 분석(②, ③, ④, ⑤): 359,195,100원(기존 - 개선 후)

구 분	기 존	개선 후
평균 월급(계급) × 인원	4,035,900(소령 7호봉) × 3명	4,035,900(소령 7호봉) × 1명
작업 소요기간('23년 작업 시)	30개월	1개월
합 계	363,231,000원	4,035,900원

- 전시에 필요한 자원소요를 정확하게 예측함으로써 전투력 유지, 신속한 대응 가능

5) 관계형 데이터베이스: 엑셀과 같은 테이블 형태로 구조화하여 저장하고 테이블 간의 관계를 이용하여 데이터를 관리
6) 벡터 데이터베이스: 데이터를 고차원 벡터로 저장해 유사성 검색을 효율적으로 수행하여 AI의 작업에 최적