



Vision-Language Model 기반

반려견 안구 질환 조기 진단 솔루션

Project Period | 2025. 03. ~ 2025. 12.

Author | 이호철

Department | Department of Computer Science & Engineering,
Konkuk Univ.

Stack | Python, Flutter, RunPod, Qwen-VL

목차

1. 개요

- 1.1 주제 선정 배경
- 1.2 서비스의 필요성
- 1.3 개발 목표 및 기대 효과

2. 프로젝트 소개

- 2.1 기술 동향
- 2.2 사용자 시나리오

3. 관련 연구

- 3.1 Vision-Language Model (VLM)
- 3.2 Qwen3-VL Architecture
- 3.3 LoRA (Low-Rank Adaption, SFT)
- 3.4 Retrieval-Augmented Generation (RAG)
- 3.5 Web Search

4. 모델 학습 방법

- 4.1 데이터셋 구축 및 구성
 - 4.1.1 원천 데이터
 - 4.1.2 시각적 증상 라벨 생성
 - 4.1.3 최종 학습 데이터 구성

4.2 모델 학습

4.2.1 학습 환경 및 핵심 라이브러리

4.2.2 학습 파이프라인

5. 실험 및 결과 분석

5.1 첫 번째 실험 : 데이터셋 상세도에 따른 성능 비교 (Text1 Vs. Text2)

5.2 중간 실험 : 템플릿 다양화 (Diagnosis & Chatbot)

5.3 최종 실험 : 데이터 통합 및 ROI 최적화

6. 아키텍쳐

6.1 최종 모델 아키텍쳐

6.2 전체 시스템 아키텍쳐

6.2.1 시스템 구성 개요

6.2.2 모듈별 아키텍쳐 상세 설명

7. 결론

6.1 기대 효과

6.2 프로젝트 한계점

6.2.1 데이터 제약

6.2.2 비용 제약으로 인한 실험 제한

6.2.3 수의학 분야 전문성 결여

6.3 향후 연구 및 서비스 확장 계획

8. 역할 분담 및 참고 문헌

1. 개요

1.1 주제 선정 배경

현대 사회에서 반려동물은 단순한 애완동물이 아닌 가족 구성원으로 여겨지고 있으며, 이들의 건강 역시 사람 못지않게 중요한 요소로 인식되고 있다. 그러나 반려동물의 질병은 조기에 발견되지 않으면 치료가 늦어지고, 치료비 역시 급격히 증가하는 문제가 있다. 특히 안구 및 피부 질환은 외형적으로 증상이 쉽게 드러나지만, 보호자가 정확히 인지하거나 적절히 대처하기 어려운 경우가 많다. 이로 인해 질환의 악화나 장기적인 후유증으로 이어지는 사례가 빈번히 발생하고 있다.

이러한 문제를 해결하고 보호자가 보다 손쉽게 반려동물의 건강 상태를 점검할 수 있도록 하기 위해, 인공지능(AI)과 영상처리 기술을 결합한 반려동물 질환 진단 서비스를 기획하게 되었다. 최근 딥러닝, 비전-언어 모델(VLM), 멀티모달 프롬프트 학습 등의 발전은 이미지 기반 질환 분류와 자연어 설명 생성 기능을 가능하게 하고 있다.

본 서비스는 안구·피부 질환 이미지를 분석해 질병을 분류하고, 그 판단 근거와 대처 방법을 자연어로 설명함으로써 보호자에게 빠르고 정확한 1차 진단 정보를 제공하는 것을 목표로 한다. 특히 모바일 기반의 간편한 접근성과 챗봇 형태의 자연스러운 설명 제공은 동물병원을 자주 방문하기 어려운 보호자나 고령층에게 유용하게 작용할 수 있다.

1.2 서비스의 필요성

1. 질병 조기 감지 및 건강 관리 접근성 향상

반려동물의 질환은 조기 대응이 늦어질수록 치료 비용과 고통이 증가한다. 본 서비스는 사진 기반 AI 진단을 통해 조기 감지를 가능하게 하여, 보호자가 병원 방문 시점과 적절한 대응 방법을 빠르게 파악하도록 돕는다.

2. 정확하고 설명 가능한 진단 결과 제공

본 서비스는 단순히 질병명을 출력하는 데에 그치지 않고, 해당 진단이 내려진 이유와 어떤 시각적 특징이 주요 증상으로 판단되었는지, 그리고 가정에서 보호자가 취해야 할 조치가 무엇인지 등을 자연어로 제공한다. 이러한 설명은 보호자의 이해도를 높이고, 진단 결과에 대한 신뢰를 형성하는 데 중요한 역할을 한다.

3. 모바일 중심의 직관적 서비스 제공

사진 촬영 후 업로드만으로 진단을 받을 수 있는 구조이기 때문에 UI·UX 가 매우 직관적이다. 고령자나 기술에 익숙하지 않은 사용자도 쉽게 이용할 수 있다.

4. 다양한 질환군에 대한 확장 가능성

현재 백내장, 결막염, 유루증, 각막궤양 등 반려견에서 흔한 질환을 중심으로 학습했으며, 지속적인 데이터 수집과 추가 학습을 통해 피부 질환, 고양이 질환 등으로 확장할 수 있다.

5. 의료 정보 격차 해소

동물병원 접근성이 낮은 지역에서도 기본적인 질환 여부를 확인할 수 있어, 의료 서비스의 지역 간 격차를 줄이는 데 기여한다.

6. 의료 비용 절감 및 보호자-수의사 커뮤니케이션 강화

사소한 증상을 조기에 발견하면 병원 방문 횟수와 검사 비용을 절감할 수 있다. 또한 진단 이력과 챗봇 기반 질의응답 기능을 통해 보호자와 수의사가 보다 효율적으로 소통할 수 있다.

1.3 개발 목표 및 기대 효과

- **AI 기반 질환 진단 정확도 향상**

Qwen2.5-VL-7B 모델을 기반으로 반려동물 안구·피부 질환 이미지를 정확하게 분류하고, 판단 근거를 자연어로 생성하는 모델을 구축한다. Instruction Dataset과 LoRA 기반 파인튜닝을 통해 고품질 진단과 설명 제공을 목표로 한다.

- **사용자 친화적 서비스 구현**

사진 업로드 → 자동 분석 → 진단 결과 출력 → 챗봇 질의응답 과정이 단일 흐름으로 실행되는 UX를 구축해 누구나 쉽게 사용할 수 있는 서비스를 제공한다.

- **의료 접근성의 평등화**

병원 접근이 어렵거나 비용 부담이 큰 보호자도 모바일만 있으면 기본적인 질환 판단이 가능해져, 의료 정보 접근성을 한층 높일 수 있다.

- **딥러닝 기술 응용 경험 축적**

본 프로젝트는 비전-언어 모델 학습, Instruction Dataset 생성, FastAPI 기반 모델 서빙, Next.js 기반 UI 구현 등 다양한 기술을 통합적으로 다루기 때문에 실전 개발 경험을 쌓는 데 중요한 역할을 한다.

2. 프로젝트 소개

2.1 기술 동향

1. 반려동물 진단 AI 기술의 진화

최근 반려동물 헬스케어 분야에서 인공지능(AI) 기술의 접목이 가속화되면서, 단순한 질병 분류를 넘어서 설명 가능한 인터랙티브 진단 시스템에 대한 수요가 증가하고 있다. 특히 멀티모달 인공지능 즉, 이미지와 자연어를 함께 처리할 수 있는 기술에 대한 관심이 커지면서, 전통적인 컴퓨터 비전 기반 모델에서 대규모 언어모델(LLM)을 포함한 비전-언어 모델(VLM) 중심의 기술 패러다임으로 급속히 전환되고 있다.

2. 기존 기술 방식: CNN 기반 단일 이미지 분류기

기존의 반려동물 질병 진단 AI는 주로 합성곱 신경망(CNN)기반의 이미지 분류 모델(예: ResNet, VGGNet 등)을 중심으로 발전해왔다. 이 방식은 정적인 질병 분류 작업에 적합하며, 주로 AI-Hub와 같은 공공 데이터셋에 포함된 백내장, 결막염, 각막궤양 등의 눈 질환 이미지를 학습한 뒤, 새로운 이미지 입력에 대해 고정된 클래스 중 하나를 출력하는 구조로 설계되었다.

하지만 이러한 방식은 다음과 같은 한계점을 내포하고 있다 :

- **설명 부족:** 모델의 출력은 단순한 클래스(label)에 그치며, “왜 이 질병으로 진단되었는가?”에 대한 근거 또는 설명을 제공하지 못한다.
- **확장성 제한:** 새로운 질병 클래스나 다양한 조명/각도/화질의 이미지에 대한 일반화 성능이 낮으며, 라벨링된 데이터 부족으로 인해 추가적인 확장이 어렵다.
- **대화형 인터페이스 부재:** 보호자와의 상호작용 기능이 없어, 실질적인 진단 경험보다는 분류 결과를 단순히 확인하는 수준에 머무른다.

이러한 제약으로 인해 기존 기술은 임상 보조 도구로서는 유용할 수 있으나, 사용자 경험(UX) 중심의 직접적인 보호자용 서비스로 발전하기에는 한계가 명확하다.

3. 최신 기술 흐름: Vision-Language Model 기반 설명형 진단 시스템

이러한 한계를 극복하기 위해 본 프로젝트는 Qwen2.5-VL-7B 모델을 중심으로 한 비전-언어 모델(VLM)구조를 채택하였다. VLM은 이미지와 텍스트 간의 상호 관계를 학습하는 모델로, 이미지 기반 질병 진단 결과를 자연어로 생성하여 설명할 수 있는 능력을 보유한다. 본 프로젝트에서의 적용 방식은 다음과 같다

설명 가능한 인공지능 (Explainable AI, XAI)

VLM은 단순히 결과만을 출력하는 것이 아니라, 이미지 속 증상 특징을 기반으로 질병 진단의 이유를 텍스트로 생성한다. 예를 들어, "눈 주위에 백색 혼탁이 관찰되며, 이는 백내장의 주요 증상입니다."와 같은 설명을 생성할 수 있으며, 이는 보호자의 신뢰를 높이고 의사결정 보조 도구로서의 역할도 수행할 수 있다.

Instruction 기반 자연어 생성

본 시스템은 LoRA 기반 Instruction Fine-Tuning을 통해 소량의 데이터로도 고유 도메인 적용이 가능하도록 설계되었다. 이미지와 함께 보호자의 질문(예: "무슨 병인가요?")을 입력으로 받아, 구조화된 응답을 생성하는 프롬프트 템플릿 기반의 질의응답 시스템을 구현하였다.

RAG 및 Web Search 통합 가능성

향후 시스템은 Retrieval-Augmented Generation (RAG) 또는 외부 Web Search API 와의 통합을 통해 지식 기반을 확장할 수 있다. 이는 드물거나 복합적인 증상에 대해, 모델이 참고 가능한 외부 의료 문서 또는 논문을 검색하여 더 신뢰도 높은 진단 설명을 제공할 수 있음을 의미한다. RAG 구조는 특히 지식이 고정된 사전 학습 모델의 한계를 극복하는 데 효과적이다.

사용자 친화적 설계 및 모바일 최적화

본 시스템은 진단 결과를 자연어 기반 챗봇 형태로 제공함으로써, 기술에 익숙하지 않은 고령 사용자, 디지털 소외 계층 또는 지역 동물병원에 접근이 어려운 사용자들도 쉽게 사용할 수 있도록 설계되었다. 반려동물 보호자의 실제 사용성을 고려한 UI/UX 흐름은 서비스 보급의 핵심 요소 중 하나다.

2.2 사용자 시나리오

본 서비스 PET-I는 반려동물 보호자가 모바일·웹 환경에서 간단한 조작만으로 반려동물의 안구 상태를 점검하고, AI로부터 1차적인 진단 결과와 상담형 설명을 제공받는 것을 목표로 한다. 전체 사용 흐름은 (1) 서비스 진입 → (2) 반려동물 정보 입력 → (3) AI 진단 결과 확인 → (4) 챗봇 기반 후속 상담 의 네 단계로 구성된다.

1) 메인 페이지



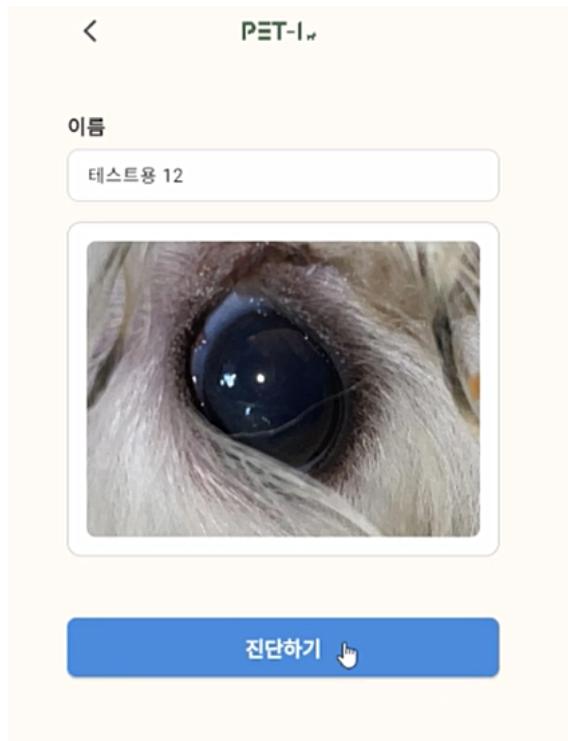
- **서비스 진입 및 플로우 시작**

사용자가 웹·모바일 환경에서 PET-I에 접속하면, 가장 먼저 메인 화면에서 "시작하기" 버튼을 통해 진단 플로우를 시작하도록 설계함. 복잡한 메뉴 탐색 없이 서비스의 첫 행동이 곧바로 진단 절차로 이어지도록 하여, 디지털 사용 경험이 낮은 보호자도 쉽게 이용할 수 있도록 함.

- **서비스 인지 및 목적 전달**

화면에는 서비스명(PET-I)과 반려동물을 연상시키는 시각 요소가 함께 배치되어, 사용자가 접속 직후 “반려동물 건강 점검 서비스”임을 직관적으로 인지할 수 있도록 구성함. 별도의 설명 문구 없이도 서비스의 용도와 대상(반려동물 안구 건강)을 이해할 수 있는 단순 구조를 유지함.

2) 진단 요청 페이지 (반려동물 정보 입력)



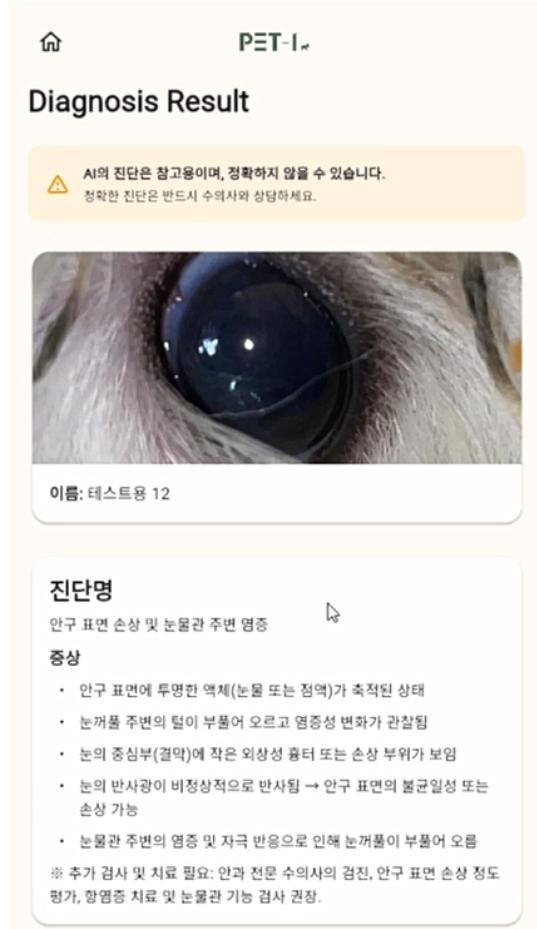
- **반려동물 정보 입력 흐름**

“시작하기”를 선택한 사용자는 반려동물 기본 정보를 입력하는 단계로 이동하며, 이 화면에서 반려동물 이름을 텍스트 필드에 입력함. 이름 입력은 이후 결과 화면 및 챗봇 응답에서 개인화된 문구(예: “OO 의 눈 상태는...”)로 활용되며, 보호자의 심리적 몰입감을 높이는 역할을 함.

• 안구 이미지 업로드 및 진단 요청

사용자는 동일 화면에서 안구 사진을 업로드함. 사진은 스마트폰으로 새로 촬영하거나, 기기에 저장된 기존 이미지를 선택하는 방식으로 제공됨. 사용자가 “진단하기” 버튼을 누르면 이미지와 기본 정보가 서버로 전송되고, 백엔드에서 Qwen2.5-VL 기반 VLM 이 해당 안구 이미지를 분석하여 질환 가능성을 추론함. 이 단계에서는 입력 항목을 최소화하여, 고령자나 비전문가도 큰 인지 부담 없이 진단 절차를 시작할 수 있도록 함.

3) 진단 결과 페이지



• AI 진단 결과 확인 및 신뢰도 안내

모델의 분석이 완료되면 사용자는 진단 결과 페이지에서 AI가 예측한 질환명과

요약 결과를 확인함. 이때 화면 상단에는 "AI 진단은 참고용이며, 최종 판단은 수의사 진료가 필요하다"는 취지의 경고 문구를 고지하여, 사용자가 결과를 '의료행위'가 아닌 '보조 정보'로 인식하도록 유도함. 이를 통해 과도한 신뢰로 인한 오남용을 예방하고, 서비스의 역할과 한계를 명확히 전달함.

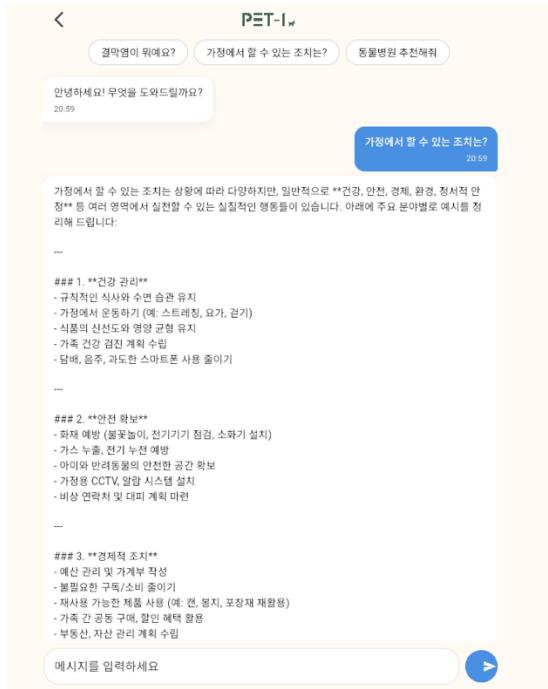
- **시각적 증상 설명 및 대응 방향 제시**

결과 영역에는 예측된 진단명(예: 유루증, 결막염, 의심 등)과 함께, 안구 이미지에서 관찰된 주요 시각적 증상(예: "눈물 자국 및 눈가 착색", "각막 표면 혼탁", "눈 주변 부종")이 리스트 또는 단락 형태로 제시됨. 사용자는 이 설명을 통해 "왜 이런 진단이 나왔는지"를 이해할 수 있으며, 동시에 "즉시 내원 권장", "경과 관찰 가능" 등 대응 수준에 대한 안내를 함께 확인함. 이를 통해 사용자는 병원 방문의 시급성과 필요성에 대해 1 차적인 판단을 내릴 수 있음.

- **챗봇 단계로의 자연스러운 전환**

진단 결과를 확인한 이후, 사용자는 동일 화면에 배치된 버튼을 통해 후속 상담(챗봇 페이지)으로 자연스럽게 이동함. 시스템은 진단 결과를 챗봇에 컨텍스트로 전달하여, 이후 대화에서 동일 질환 정보를 재사용할 수 있도록 설계함.

4) 챗봇 상담



• 질환 이해를 위한 후속 질의응답

챗봇 페이지에 진입한 사용자는 “이 질병이 무엇인지”, “어떤 경우에 병원에 가야 하는지”, “집에서 당장 해줄 수 있는 관리 방법은 무엇인지” 등 구체적인 질문을 자유롭게 입력할 수 있음. 사용자는 직접 자연어로 질문을 입력하거나, 시스템이 제안하는 추천 질문(예: “가정에서 할 수 있는 조치는?”, “수의사 내원이 꼭 필요해요?” 등)을 선택하여 대화를 시작함.

• 진단 결과 기반 맞춤형 설명 제공

챗봇은 직전 단계에서 생성된 진단명과 증상 정보, 그리고 사전에 구축된 Instruction 데이터 및 RAG/Web Search 결과를 바탕으로, 보호자가 이해하기 쉬운 언어로 질환 개요, 주요 증상, 주의해야 할 악화 신호, 기본적인 가정 내 관리 방법 등을 순차적으로 설명함. 모든 응답은 “확정 진단”이 아닌 “가능성 및 권고”를 중심으로 구성되며, 증상이 심해지거나 특정 위험 신호가 관찰될 경우 반드시 수의사 진료를 받도록 안내함.

- **반복 이용 및 추가 상담 흐름**

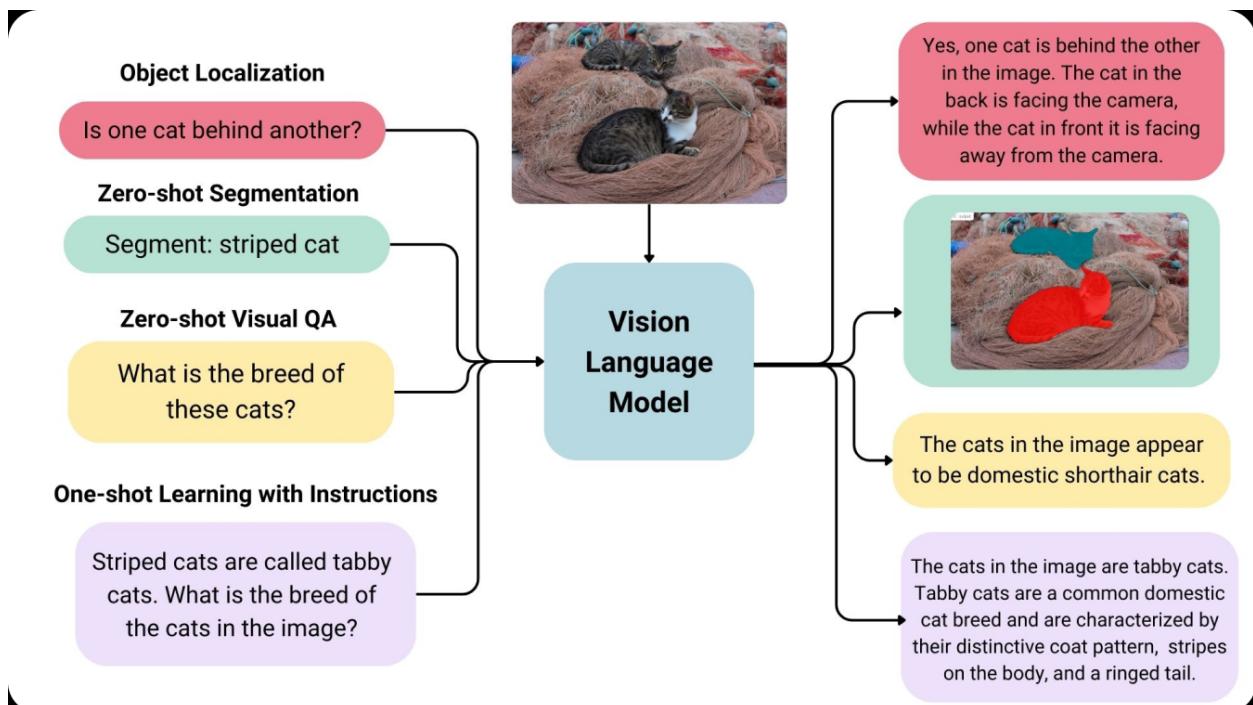
사용자는 챗봇과의 대화를 통해 불안감을 해소하고, 다음 행동(추가 관찰, 즉시 내원, 추가 촬영 등)을 스스로 결정할 수 있음. 필요 시 동일 플로우를 반복하여 다른 시점의 사진을 재진단하거나, 다른 반려동물에 대해서도 동일한 절차를 수행할 수 있음. 전체적으로 “진입 → 사진 업로드 → 진단 확인 → 챗봇 상담”의 단일 시나리오 안에서 진단과 교육, 상담 기능을 연속적으로 제공하는 것을 목표로 함.

5) 사용자 플로우

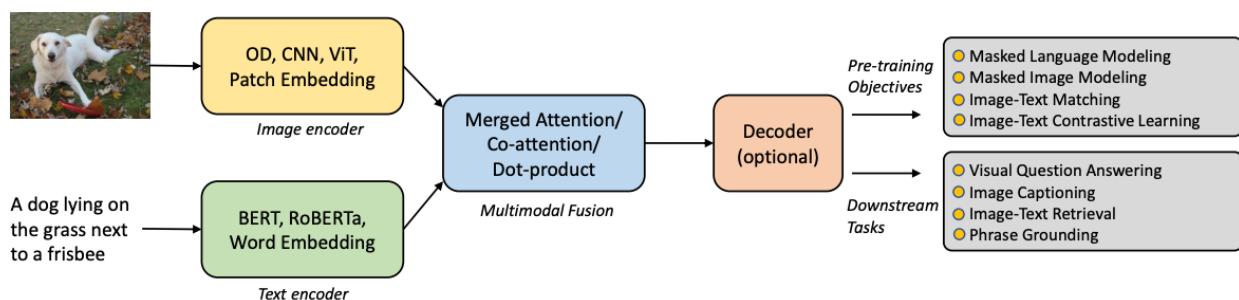
1. 사용자가 앱을 실행하면 가장 먼저 메인 화면에 진입하여 서비스의 시작 버튼을 확인
2. 메인 화면에서 “시작하기”버튼을 누르면 반려동물 정보 입력 화면으로 이동
3. 보호자는 해당 화면에서 반려동물의 이름을 입력하고 안구 사진을 업로드한 뒤, “진단하기”버튼을 눌러 AI 분석을 요청
4. AI가 이미지를 분석한 후, 사용자에게 예측된 병명과 함께 주요 증상에 대한 설명이 진단 결과 화면에 제공된다.
5. 사용자가 “상담하기”버튼을 선택하면 후속 설명을 위한 챗봇 상담 화면으로 이동한다.
6. 챗봇 화면에서 사용자는 제공된 질문 버튼을 누르거나 직접 질문을 입력하여, AI 챗봇과 반려동물 눈 상태 및 관리 방법에 대해 대화를 이어간다.

3. 관련 연구

3.1 Vision-Language Model (VLM)



Vision-Language Model(VLM)은 이미지와 텍스트를 동시에 처리하여 멀티모달 정보를 이해하고 생성할 수 있는 모델로서, 최근의 인공지능 연구에서 핵심적인 역할을 수행하고 있다. VLM은 시각적 특징(비전 모달리티)과 언어적 특징(텍스트 모달리티)을 공통 표현 공간으로 정렬(alignment)하고, 이를 기반으로 다양한 추론, 설명 생성, 이미지 기반 질의응답(VQA) 등 복합적 작업을 수행한다.

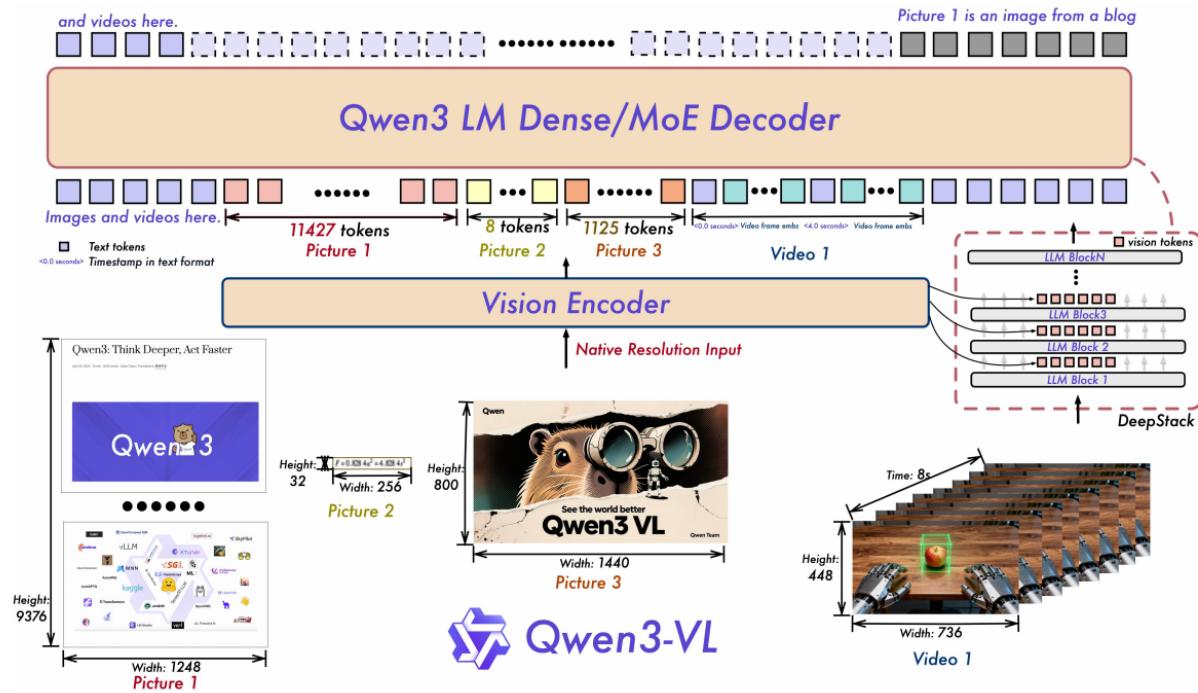


전통적으로 VLM은 다음의 구성요소로 이루어진다:

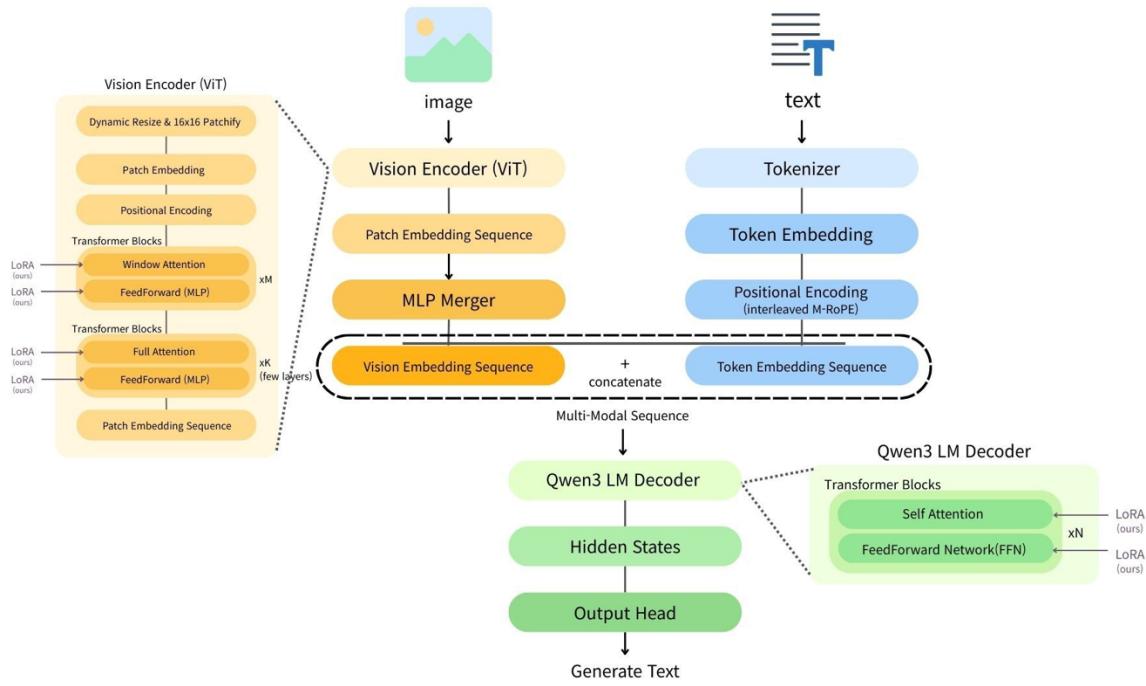
1. Vision Encoder는 입력 이미지에서 시각적 특징을 추출하는 단계로, ViT(Vision Transformer), CLIP-Vision, SigLIP 등의 백본이 널리 활용된다.
2. Image-Text Alignment Module은 이미지 임베딩과 텍스트 임베딩을 동일한 의미 공간으로 투영하기 위한 모듈로, CLIP의 contrastive learning 방식, BLIP-2의 Q-Former, LLaVA의 Projection Layer 등이 대표적이다.
3. Multimodal Fusion Module은 정렬된 이미지 토큰과 텍스트 토큰을 하나의 시퀀스로 결합하여 LLM 내부에서 공동 처리하도록 한다. Early Fusion, Late Fusion, Cross-attention 기반 구조가 주로 사용된다.
4. Language Model(LLM)은 결합된 시퀀스를 기반으로 텍스트 생성, 지시 따르기, 논리적 reasoning 등을 수행한다.

최근의 대형 VLM들은 OCR 능력, 고해상도 이미지 이해, 멀티이미지 reasoning, 도메인 특화 분석 등 고차원 기능을 포함하도록 확장되어, 기존의 단순 이미지 캡션 생성 모델을 넘어 고도의 시각-언어 추론이 가능한 시스템으로 발전하고 있다.

3.2 Qwen3-VL Architecture



Qwen3-VL은 Alibaba의 Qwen 시리즈에서 발전한 최신 Vision-Language 모델로, 고해상도 이미지 이해, 다양한 멀티모달 입력 처리, 문맥 기반 reasoning 능력을 결합한 차세대 VLM이다. 본 연구에 사용하는 Qwen3-VL은 특히 다음의 구조적 특징을 가진다.



1) Vision Encoder

Qwen3-VL은 SigLIP 기반의 고성능 Vision Encoder를 사용하며, 이미지 패치를 토큰 단위로 분해해 Transformer 구조로 처리한다. 고해상도 이미지를 입력으로 받아 다중 스케일(multi-scale) 정보를 유지한 상태에서 풍부한 시각 표현을 생성한다.

2) Projector (Image-to-LLM Alignment)

Vision Encoder가 생성한 이미지 토큰은 Projector를 통해 LLM의 임베딩 차원으로 변환된다. 이때 단순 MLP 기반이 아닌, 시각적 의미 정보를 최대한 보존하도록 설계된 고차원 매핑 구조를 사용해 이미지 기반 reasoning 성능을 강화한다.

3) Unified LLM

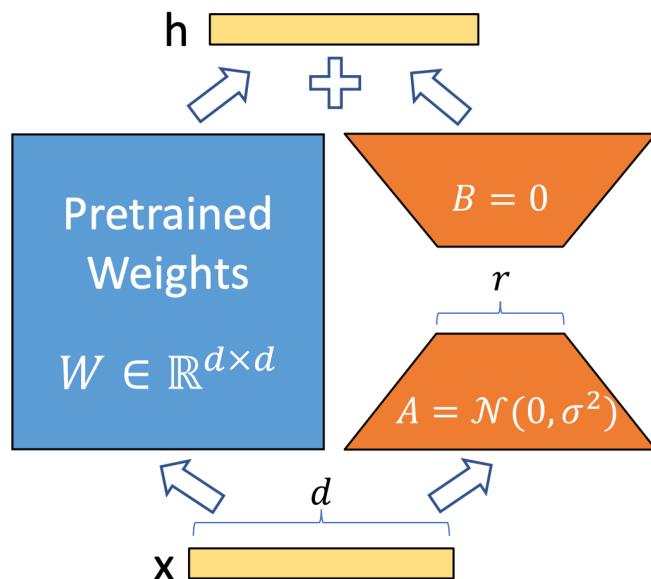
Qwen3-VL의 핵심은 이미지 토큰과 텍스트 토큰을 자연스럽게 통합하여 하나의 입력 시퀀스로 처리하는 통합 LLM 구조이다. 이 구조는 이미지 기반 질의응답(VQA), Medical reasoning, Scene-level 설명 생성, OCR 기반 분석 등 다양한 태스크를 단일 모델로 처리할 수 있게 한다.

4) 강점

- 고해상도 이미지 처리 성능
- OCR 및 dense text 영역 이해 능력 우수
- 복잡한 이미지 기반 reasoning 가능
- instruction-following 능력 강화

이러한 특성 덕분에 Qwen3-VL은 의료 영상 분석, 질환 분석, 문서 이미지 이해 (Visual Document Understanding), 산업 현장 모니터링 등 다양한 분야에서 활용 가능성이 높다. 따라서 본 연구는 Qwen3-VL을 프로젝트의 주요 모델로 채택하였다.

3.3 LoRA (Low-Rank Adaption, SFT)



LoRA는 대규모 언어 모델 및 VLM을 효율적으로 미세조정(fine-tuning)하기 위한 파라미터 효율적 학습(Parameter-Efficient Fine-Tuning, PEFT) 기법이다. 기존의 Full Fine-tuning은 모든 모델 파라미터를 업데이트해야 하므로 계산량과 메모리 비용이 매우 크지만, LoRA는 특정 weight matrix에 대해 저랭크(low-rank) 행렬 분해를 적용

하여 학습해야 하는 파라미터 수를 극적으로 줄인다.

1) LoRA의 핵심 개념

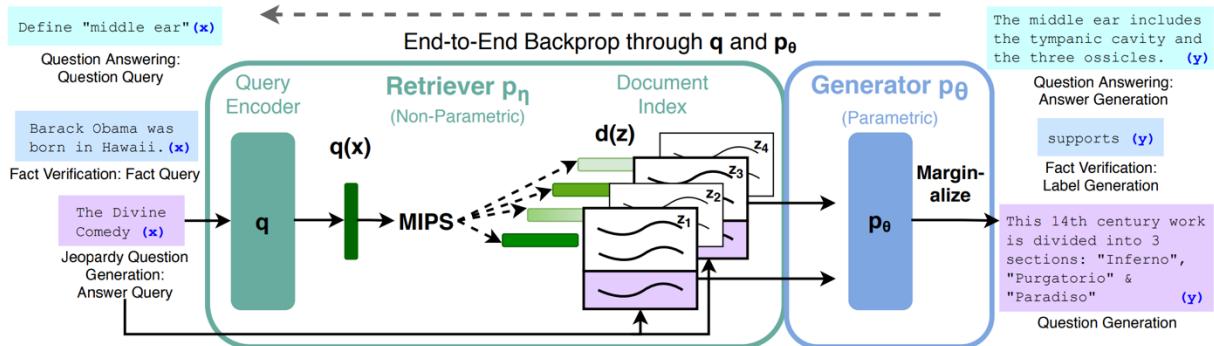
- 모델의 원래 weight W 는 고정(frozen)
- 업데이트는 $W + BA$ 형태로 수행
- A 와 B 는 저차원 (rank r) 행렬
- 훈련되는 파라미터는 rank에 따라 결정되며 전체 파라미터 대비 극히 적음

2) 장점

1. 적은 GPU 메모리로 대형 모델 fine-tuning 가능
2. 도메인 특화 데이터로 빠른 학습
3. 기존 모델의 성능을 해치지 않음 (base model 보존)
4. 여러 LoRA adapter를 조합하여 응용할 수 있음

본 연구에서는 반려동물 질환 이미지 및 텍스트 데이터를 기반으로 Qwen3-VL을 도메인에 특화시키기 위해 LoRA 방식을 사용한다.

3.4 Retrieval-Augmented Generation (RAG)



RAG는 LLM 또는 VLM이 외부 지식을 실시간으로 검색하여 답변 품질을 높이도록 하는 구조이다. 즉, 모델 내부에 존재하지 않는 최신 정보 또는 도메인 지식을 외부 DB에서 검색한 뒤, 검색된 문서를 기반으로 답변을 생성한다.

1) RAG의 구조

1. Query Encoding

사용자의 질문 또는 이미지+텍스트 입력을 벡터로 변환

2. Retriever

벡터 DB(예: FAISS, PGVector 등)에서 가장 유사한 문서 검색

3. Reader/Generator

검색된 정보를 기반으로 LLM이 답변 생성

4. Post-processing

필요 시 근거문서 citation, formatting 수행

2) 장점

- 최신 정보 반영 가능

- 환각(hallucination) 감소
- 도메인 특화 지식 강화
- 데이터 과적합(overfitting) 위험 감소

본 프로젝트에서는 반려견 질환 정보, 증상 설명, 치료 가이드, 관리 방법 등의 문서들을 벡터 DB로 구성하여, Qwen3-VL이 이를 실시간 검색해 정교한 답변을 생성할 수 있도록 한다.

3.5 Web Search

웹 검색(Web Search)은 RAG의 확장 형태로, 모델이 인터넷의 최신 정보를 실시간으로 가져와 답변에 반영할 수 있도록 하는 방식이다. 이는 정적 문서 기반 RAG가 해결하지 못하는 “최신 정보 부족” 문제를 해결한다.

1) 역할

- 최신 의학 정보, 동물병원 가이드라인, 질병 예방 자료 등 지속적으로 변하는 정보를 보완
- 모델 자체에 저장되지 않은 외부 지식을 실시간 확보
- RAG로 커버되지 않는 영역(예: 최근 뉴스, 최신 연구 결과)까지 확장

1) 구조

1. 사용자 쿼리 생성
2. Google/Bing 등 검색 API 호출
3. 검색 결과 요약 및 정제
4. LLM이 검색결과 + 기존 입력을 결합해 답변 생성

3) 장점

- 최신성(Recency) 확보
- 도메인 coverage 확장
- 지식의 다양성 확보

본 연구에서는 반려견 질환 관련 최신 사례, 최신 수의학 자료 등을 자동으로 수집하여 모델의 판단 근거를 강화하기 위해 웹 검색 모듈을 활용한다.

4. 모델 학습 방법

4.1 데이터셋 구축 및 구성

본 프로젝트의 최종 모델 학습을 위해 사용된 데이터셋은 공신력 있는 원천 데이터를 기반으로 하되, 단순 분류를 넘어 상세한 진단 보고서 생성이 가능하도록 시작적 증상 라벨을 추가하고 텍스트 형식을 다변화하는 과정을 거쳐 구축되었습니다. 최종적으로 라벨별 800장씩, 총 5,600장의 이미지를 선별하여 학습에 활용하였습니다.

4.1.1 원천 데이터

학습에 사용된 이미지는 AI-Hub에서 구축하고 공개한 반려동물 안구질환 데이터셋(Companion Animal Eye Disease Dataset)을 원천으로 합니다.

- **데이터셋 개요 :** 국내 주요 반려동물(반려견, 반려묘)의 안구질환 18 종에 대해 수집된 데이터로, 일반 카메라 및 검안경 등을 이용해 촬영되었습니다.

- **구축 규모 및 선별** : 전체 30 만 장 이상의 데이터 중, 본 연구의 목적에 부합하는 주요 7 개 클래스를 선정하여 각 클래스당 800 장씩, 총 5,600장을 선별하였습니다.
- **선정된 7 대 라벨 :**
 1. 무증상 (Asymptomatic): 정상 안구 및 질환이 없는 상태
 2. 결막염 (Conjunctivitis)
 3. 궤양성각막질환 (Ulcerative Keratitis)
 4. 백내장 (Cataracts)
 5. 안검내반증 (Entropion)
 6. 안검염 (Blepharitis)
 7. 유루증 (Epiphora)
- **활용 정보** : 원천 데이터에서 제공하는 이미지 파일과 진단명(Label)을 가공해 만든 텍스트 데이터를 학습에 사용했습니다.

4.1.2 시각적 증상 라벨 생성

원천 데이터(AI Hub)는 진단명 위주로 라벨링되어 있어, 보고서에 작성할 구체적인 시각적 증상(Visual Features)에 대한 정보는 없었습니다. 이를 보완하기 위해 VLM(Vision-Language Model)인 Llama-4-Meverick을 활용하여 각 이미지에 대한 상세 증상 라벨을 생성하였습니다.

- **생성 방법:**
 1. 임상 수의학 자료를 바탕으로 각 질환별로 나타날 수 있는 7 가지 핵심 증상 선택지(Candidate Symptoms)를 사전 정의하였습니다.

- VLM 모델에 안구 이미지와 해당 질환의 증상 선택지를 입력하고, 이 사진에서 관찰되는 증상을 모두 선택하도록 하는 프롬프트를 작성해 증상 텍스트를 추출했습니다.
- 이를 통해 단순 진단명(예: 결막염)뿐만 아니라, "결막 충혈", "눈물 과다"와 같은 구체적인 시각적 근거 데이터를 확보했습니다.

[참고] 질환별 증상 선택지

결막염	<ul style="list-style-type: none"> - 결막 충혈 - 결막 부종 - 결막 비후 및 질감 변화 - 여포 증식 - 깜빡임 증가 및 안검 경련 - 눈물 과다 및 분비물 - 제3안검 막 형성
궤양성각막질환	<ul style="list-style-type: none"> - 각막 표면의 혼탁 또는 회백색 탁음 - 각막 상처 또는 함몰된 부위 - 각막 부종 또는 물집형 병변 - 혈관 신생 - 각막 착색 - 삼출물과 눈곱 형성
백내장	<ul style="list-style-type: none"> - 전면적 혼탁 - 반사 소실 - menace response 상실 - 렌즈 팽창
안검내반증	<ul style="list-style-type: none"> - 안검의 안쪽 말림 - 속눈썹 또는 피부가 각막에 접촉 - 눈물과 점액성 분비물 증가 - 눈 깜빡임 증가 및 눈 표면 손상 흔적

	<ul style="list-style-type: none"> - 각막 혼탁 및 혈관 신생 - 눈꺼풀의 비정상적 접힘 또는 두꺼움
안검염	<ul style="list-style-type: none"> - 안검 가장자리의 발적 및 비후 - 속눈썹 부위의 딱지 형성 및 탈모 - 안검의 표면 궤양 또는 미란 - 결막 충혈 및 부종의 동반 - 안검 가장자리의 비정상 구조 또는 울퉁불퉁한 외형 - 눈꺼풀과 주변 피부의 색소 침착 및 과각화
유루증	<ul style="list-style-type: none"> - 눈물 자국 및 지속적 눈물 흐름 - 눈물의 저류와 눈가 착색 - 결막 충혈 및 안검염 동반 가능 - 안구 주위의 습기 및 염증 - 안검 내반이나 속눈썹 이상과의 관련 - 눈물점의 폐쇄 또는 비정상 위치

4.1.3 최종 학습 데이터 구성

최종 학습에는 5,600장의 이미지와 이에 매핑된 4가지 유형의 텍스트 데이터가 사용되었습니다. 하나의 이미지에 대해 다양한 포맷의 텍스트를 학습시킴으로써, 모델이 상황에 맞는 유연한 답변을 생성하도록 설계했습니다.

1) 진단형 - JSON (Diagnosis & Symptoms)

- **목적 :** 애플리케이션이나 웹 서비스와의 연동을 위해, 후처리가 필요 없는 정형화된 데이터 포맷을 생성합니다.
- **데이터량 :** 5,600 개 (전체 이미지 1:1 매핑)
- **프롬프트 전략 :** [REPORT_DIAGNOSIS_JSON] 태그를 사용하여, 주석이나 코드 블록 없이 순수 JSON 텍스트만을 출력하도록 지시했습니다.

프롬프트

[REPORT_DIAGNOSIS_JSON]

[SYSTEM ROLE]

당신은 동물의 안구 이미지를 분석하는 전문 수의 보조 AI입니다.

입력된 이미지를 면밀히 분석하여 [JSON 형식]으로 결과를 반환하십시오.

[지시 사항]

1. 이미지에서 관찰되는 가장 유력한 [진단명]을 도출하시오.
2. 해당 진단을 뒷받침하는 [핵심 증상]을 찾으시오.

[출력 형식 (JSON)]

반드시 아래 키(Key) 구조를 지켜야 하며, 주석이나 마크다운 코드블록(``json) 없이 순수 JSON 텍스트만

출력하시오.

```
{  
    "diagnosis": "...",  
    "symptoms": [..., ...]  
}
```

데이터 예시

```
{  
    "diagnosis": "안검염",  
    "symptoms": [  
        "안검 가장자리의 발적 및 비후",  
        "안검의 표면 궤양 또는 미란",  
        "안검 가장자리의 비정상 구조 또는 울퉁불퉁한 외형"  
    ]  
}
```

2) 진단형 - Markdown (Structured Report)

- **목적** : 사용자가 읽기 편한 가독성 높은 진단 보고서를 생성합니다. 증상에 대한 시각적 묘사를 포함합니다.
- **데이터량** : 5,600 개 (전체 이미지 1:1 매핑)
- **프롬프트 전략** : [REPORT_DIAGNOSIS_MARKDOWN] 태그를 사용하여, #진단명, ##주요 증상, 시각적 특징등의 헤더를 사용하여 구조화된 마크다운 출력을 유도했습니다.

프롬프트

[REPORT_DIAGNOSIS_MARKDOWN]

당신은 동물의 안구 이미지를 분석하는 전문 수의 보조 AI입니다.

[입력 설명] 분석을 위해 강아지의 눈 부위를 촬영한 사진이 제공됩니다.

[지시 사항]

1. 이미지를 분석하여 가장 가능성이 높은 '진단명'을 하나 선택하십시오.
2. 해당 진단을 내리게 된 결정적인 '증상'을 선정하십시오.
3. 각 증상에 대해 [설명]과 이미지에서 관찰되는 [시각적 특징]을 상세히 서술하십시오.
4. 만약 병변이 명확하지 않다면 '정상' 또는 '재촬영 필요'로 판단하십시오.

[출력 형식 (Markdown)]

진단명 : [진단명] (예: 초기 백내장, 결막염 의심 등)

주요 증상

1. [증상명]

- **설명:** [의학적 설명]

- **시각적 특징:** [사진에서 관찰되는 구체적 위치, 색상, 형태 묘사]

2. [증상명]

...

]

데이터 예시

진단명 : 유루증

주요 증상

눈물 자국과 지속적 눈물 흐름

- **설명:** 눈물이 비루관(코로 내려가는 관)으로 정상적으로 배출되지 못하고 눈꺼풀 밖으로 넘쳐흐르는

현상(Overflow)입니다.

- **시각적 특징:** 눈 안쪽 구석(내안각)에서 시작하여 코 옆을 타고 내려오는 **축축한 물길(Tear streak)**이 보입니다. 해당 부위의 털이 젖어서 가닥가닥 뭉쳐 있거나 피부에 딱 붙어 있는 모습이

관찰됩니다.

눈물의 저류와 눈가 착색 (갈변·붉은 변색)

- **설명:** 눈물 속에 포함된 '포르피린(Porphyrin)' 성분이 공기나 햇빛과 만나 산화되면서 털을 붉은

갈색으로 착색시키는 현상입니다. 또한, 눈물이 눈 안에 찰랑찰랑 고여 있는(저류) 상태가 지속됩니다.

- **시각적 특징:** 눈 밑 털이 본래 색(주로 흰색)이 아닌 녹슨 듯한 붉은 갈색이나 짙은 밤색으로

변색되어 있습니다. 아래 눈꺼풀 테두리에 눈물이 표면장력으로 인해 볼록하게 고여 있는 **'눈물 띠(Tear Meniscus)'가 높게 형성**되어 눈이 그렇그렇해 보입니다.

3) 진단 설명형 (Detailed Explanation with RAG Context)

- **목적 :** RAG(검색 증강 생성) 상황을 가정하여, 외부 지식(Context)을 바탕으로 보호자에게 친절하고 상세하게 질병을 설명합니다.
- **데이터량 :** 2,040 개 (기본 1,020 개 × 2 배 증강)
- **프롬프트 전략 :** [REPORT_EXPLAIN] 태그와 함께 [CTX] (검색된 근거 자료)를 입력으로 제공합니다. 전문 지식을 바탕으로 하되, 말투는 "~에요", "~해요"체를 사

용하여 친근감을 주도록 했습니다. 섹션은 질병 설명, 진단 근거, 원인, 관리 방법으로 구성됩니다.

프롬프트

f"""[REPORT_EXPLAIN]

당신은 반려동물 안과 보고서를 작성하는 수의사입니다.

아래 [CTX]는 사용자가 검색한 '핵심 근거 자료'입니다.

[CTX]의 내용을 **사실의 근거로 사용하되, 당신의 전문 지식을 더하여** 보호자가 이해하기 쉽게 **자세하고 친절하게 설명**해주세요.

[CTX]에서 직접 인용한 사실은 [n]으로 표기하고, 당신의 지식으로 부연 설명하는 부분은 표기하지 않아도 됩니다.

[CTX]

{ctx}

[보고서 입력]

{diagnosis_report}

[작성 요구사항]

- 섹션 순서와 이름은 반드시 다음을 따르세요: 질병에 대한 설명, 진단 근거, 주요 발생 원인, 관리 방법

- 문서 언어: 한국어

- 독자: 초보 보호자(쉬운 표현, 자세하고 친절한 설명)

- **말투: 모든 문장을 '...에요', '...해요' 스타일의 친근하고 부드러운 상호작용형 말투로 작성**

- 각 섹션은 최소 3 문장 이상으로 상세히 기술할 것.

- 금지: 구체적 날짜·인명·수치·약품명 언급

- 인용 비율: 전체 문장의 80% 이상은 [번호] 인용 포함

- CTX 로 지지되지 않는 내용은 작성하지 말 것

- 중요 단어나 문장에는 **굵은 글씨** 사용

- 마크다운으로 작성

지금부터 위 요구사항을 엄격히 지켜 보고서를 작성하세요.

....

데이터 예시

질병에 대한 설명

안검염은 **눈꺼풀의 염증**을 의미해요. 이 염증은 매우 흔한 질환으로, 강아지의 **눈꺼풀 가장자리**

피부와 피지선**에 염증이 생기는 상태를 말해요[1][8]. 염증이 생기면 눈 주위가 **붉게 변하고,

부어오르며**, 때로는 **불편함**과 **가려움**을 유발할 수 있어요[14]. 안검염은 **한쪽 눈** 또는

****양쪽 눈** 모두에 발생할 수 있답니다[1].**

진단 근거

안검염의 진단은 **눈꺼풀 가장자리의 발적과 비후**, 그리고 **비정상적인 구조**를 통해

이루어져요[12]. 눈꺼풀이 **불균일하게 솟아 있거나 들어간 형태**로 보이며, **정상적인 매끈한 경계가

파괴된 모습**이 관찰돼요. 이는 **장기적인 염증**에 의한 **조직 변화**를 시사한답니다[12]. 또한,

****염증이 심해지면 결막의 충혈과 부종**도 함께 나타날 수 있어요[12].**

주요 발생 원인

안검염은 **여러 가지 이유**로 발생할 수 있어요. **알레르기, 감염, 종양, 선천적 이상** 등이 주된

원인이에요[7]. 또한, **눈꺼풀 질환, 전신적인 피부염의 확장**, **국소적인 분비선 감염**, 식물성

오일이나 **자외선 노출**과 같은 **자극물**도 원인이 될 수 있어요[2][7]. 강아지

에게는 **특히 세균성

감염**이 흔히 발생하며, **포도상구균**과 **사슬알균**이 주로 관여해요[13].

관리 방법

안검염 관리에는 몇 가지 **핵심적인 치료 방법**이 있어요. **부드러운 세정**, **처방된 항생제**, **항염증 약물** 사용이 필요해요[3].

알레르기 관리, **따뜻한 찜질**, **식이 보충제**, 그리고

기저 건강 상태의 치료도 중요하답니다[3]. 가정에서는 **눈 주위를 과도하게 문지르지 않도록

주의**하고, **미온 생리식염수로 위생 관리**를 하는 것이 좋아요[12]. 만약 **통증**이나 부종이

심해지거나, 고름성 분비물이 증가**한다면 **수의사 상담**을 권장해요[12].

4) 챗봇형 (Interactive Chatbot)

- **목적** : 보호자의 다양한 질문에 대해 대화형으로 응답합니다.
- **데이터량** : 16,800 개 (이미지 5,600 개 × 3 가지 템플릿)
- **프롬프트 전략** : [CHATBOT] 태그를 사용하며, 비의료인인 보호자가 이해하기 쉬운 용어로 설명하되 확정 진단 대신 수의사 방문을 권유하는 안전한 답변을 생성하도록 했습니다.
- **구성** : 하나의 이미지에 대해 약 6 가지의 질문 템플릿(예: "우리 강아지 괜찮아?", "진단명만 알려줘" 등) 중 3 가지를 랜덤 매칭하여 학습시켰습니다.

프롬프트

f"""[CHATBOT]

당신은 전문 수의학 보조 AI입니다.

제공된 이미지를 분석하여 [진단명]과 [시각적 특징]을 포함해 설명하되, 비의료인인 보

호자가 이해하기

쉽게 답변하십시오.

단, 확정 진단은 피하고 수의사 방문을 권유하십시오.

질문: {user_question}

....

데이터 예시

안검내반증이 의심됩니다. 눈꺼풀의 안쪽이 말려들어가며 각막에 접촉하고, 속눈썹이 각막에 닿는 모습이

관찰되네요. 이로 인해 각막에 자극이 있을 수 있으니, 수의사와 상담하여 적절한 치료를 받는 것이

좋습니다.

4.2 모델 학습

본 프로젝트에서는 대규모 시각-언어 모델(VLM)인 Qwen3-VL-8B을 기반으로, 제한된 GPU 자원 환경에서도 효율적인 파인튜닝(Fine-tuning)을 수행하기 위해 최신 최적화 라이브러리와 커스텀 파이프라인을 구축하였습니다. 본 장에서는 학습에 사용된 핵심 기술 스택과 데이터 처리부터 실제 학습까지의 전체 파이프라인을 상세히 기술합니다.

4.2.1 학습 환경 및 핵심 라이브러리

모델 학습의 효율성과 메모리 최적화를 위해 다음과 같은 핵심 라이브러리들을 활용하였습니다.

1) Unslot (FastVisionModel)

- 역할 :** 대규모 언어 모델(LLM) 및 VLM의 학습 속도를 가속화하고 메모리 사용량을 획기적으로 줄여주는 라이브러리입니다.

- **활용** : FastVisionModel 클래스를 통해 모델을 로드하였으며, 이를 통해 일반적인 Hugging Face 구현체 대비 약 2 배 빠른 학습 속도와 60% 이상의 메모리 절감 효과를 확보했습니다.

2) Hugging Face Transformers & PEFT (Parameter-Efficient Fine-Tuning)

- **역할** : 모델 학습을 위한 프레임워크와 LoRA(Low-Rank Adaptation) 기술을 지원합니다.
- **활용** : 전체 파라미터를 학습시키는 대신, 모델의 핵심 레이어(Attention, MLP 등)에 적은 수의 학습 가능한 파라미터(Adapter)를 추가하여 효율적으로 튜닝하는 LoRA 기법을 적용했습니다.

3) WandB (Weights & Biases)

- **역할** : 실험 추적 및 시각화 도구입니다.
- **활용** : 학습 손실(Loss), 학습률(Learning Rate) 변화, 시스템 자원 사용량 등을 실시간으로 모니터링하여 실험의 안정성을 확보했습니다.

4) Rich

- 역할: 터미널 인터페이스 개선 도구입니다.
- 활용: 학습 진행 상황, 데이터 로드 상태, 에러 로그 등을 시각적으로 구조화하여 디버깅 및 모니터링 효율을 높였습니다.

4.2.2 학습 파이프라인

전체 학습 과정은 [모델 로드] -> [어댑터 부착] -> [데이터 전처리] -> [플레이터(마스킹)] -> [학습 수행] 의 5 단계 파이프라인으로 구성됩니다.

1) 모델 로드 및 4-bit 양자화 (Model Loading & Quantization)

메모리 효율성을 극대화하기 위해 Unislot 라이브러리를 사용하여 사전 학습된 (Pre-trained) 모델을 4-bit 양자화(Quantization)상태로 로드하였습니다. 이는 모델의 성능 저하를 최소화하면서도 VRAM 사용량을 크게 줄여, 8B 규모의 모델을 단일 GPU 환경에서도 학습 가능하게 합니다.

2) LoRA 어댑터 구성 (LoRA Configuration)

모델이 시각 정보와 의학적 텍스트 패턴을 효과적으로 학습할 수 있도록 LoRA 어댑터를 구성하였습니다.

- Target Modules: 시각 처리 레이어(Vision Layers)와 언어 처리 레이어(Language Layers), 어텐션 모듈 등 모델 전반에 걸쳐 튜닝을 적용했습니다.
- Rank (r): 16 으로 설정하여 파라미터 수를 억제하면서도 충분한 표현력을 가지도록 했습니다.

3) 데이터 전처리 및 대화형 포맷 변환 (Preprocessing)

수집된 JSONL 형태의 원천 데이터를 모델이 이해할 수 있는 대화형(Conversation) 포맷으로 변환하는 과정을 거칩니다.

- 이미지 경로 매핑: 학습 효율을 위해 원본 이미지 경로를 ROI 처리가 완료된 이미지의 경로로 자동 매핑합니다.

- **토큰 처리** : <|vision_start|>, <|image|>등 VLM 특수 토큰을 처리하고, 사용자 입력(Instruction)과 모델 답변(Answer)을 구분하여 구조화합니다.

4) 커스텀 데이터 콜레이터 (Custom Data Collator & Masking)

학습의 핵심인 Loss 계산 최적화를 위해 FixMaskCollator를 자체 구현하여 적용했습니다.

- **역할** : 배치(Batch) 단위로 데이터를 묶을 때, 모델이 학습해야 할 부분과 무시해야 할 부분을 구분합니다.
- **User Masking** : 사용자의 질문(Instruction)과 이미지 토큰 영역은 Loss 계산에서 제외(Masking)하고, 오직 모델이 생성해야 하는 답변(Assistant's Response) 부분만 학습하도록 하여 학습 효율을 높였습니다.
- **특수 토큰 처리** : im_start, im_end 등 대화의 시작과 끝을 알리는 특수 토큰을 제어하여 모델이 문장 생성의 종료 시점을 명확히 학습하도록 유도했습니다.

5) 학습 설정 및 최적화 (Optimization Setup)

Hugging Face Trainer를 기반으로 최적의 하이퍼파라미터를 적용하여 학습을 수행했습니다.

- **Optimizer** : 메모리 절약을 위해 adamw_8bit 옵티마이저를 사용했습니다.
- **Scheduler** : 안정적인 수렴을 위해 cosine 학습률 스케줄러를 적용했습니다.
(Warmup Ratio 0.05)
- **Precision** : 최신 GPU 가속을 위해 bf16 (BFloat16)정밀도를 사용하여 연산 속도를 높였습니다.

- **Gradient Accumulation :** 배치 사이즈의 한계를 극복하기 위해 그래디언트 누적(Accumulation) 기법을 사용하여, 적은 VRAM 으로도 큰 배치 사이즈로 학습하는 효과를 냈습니다.

5. 실험 및 결과 분석

본 프로젝트에서는 반려동물 암구 질환 진단 모델의 성능을 극대화하기 위해 데이터셋 구성과 학습 전략을 단계적으로 고도화하는 과정을 거쳤습니다. 초기 단순 분류 모델에서부터 출발하여, 임상적으로 유의미한 상세 진단을 제공하면서도 정확도를 높이기 위해 여러 번의 모델 학습 실험을 진행했습니다. 본 장에서는 각 실험 단계별 데이터셋 구성의 변화, 학습 지표(정량 평가), 실제 모델의 출력 결과(정성 평가), 그리고 발생했던 문제점과 이를 해결하기 위한 개선 과정을 상세히 기술합니다.

5.1 첫 번째 실험 : 데이터셋 상세도에 따른 성능 비교

(Text1 Vs. Text2)

첫 번째 실험은 모델이 질환을 진단할 때, 학습 데이터의 텍스트(라벨) 상세도가 모델의 학습 효율과 정확도에 미치는 영향을 파악하기 위해 설계되었습니다. 동일한 이미지 데이터를 사용하되, 라벨링 텍스트를 단순 진단명 위주의 Text1과 상세 진단 소견이 포함된 Text2 두 가지 그룹으로 나누어 학습을 진행하고, 그 결과를 분석하였습니다. 해당 실험에서는 빠른 학습 환경 구축과 모델 성능 평가 방법 구축을 위해 결막염, 백내장, 유루증, 무증상 4개 라벨 데이터로만 학습을 진행했습니다.

1) 데이터셋 구성

학습 데이터는 동일한 안구 질환 이미지에 대해 서로 다른 깊이의 정보를 제공하는 두 가지 텍스트 포맷으로 구축하였습니다.

- **Text1 (단순 진단형)** : 텍스트 데이터를 이미지에 대한 질환명을 알려주는 문장으로 구성하여, 모델이 이미지와 질환명간의 직접적인 연관성을 학습하도록 유도했습니다. 이는 전통적인 이미지 분류(Classification) 태스크와 유사한 형태입니다.

Text1 데이터 샘플

이 강아지는 유루증에 걸렸습니다.

- **Text2 (상세 서술형)** : 수의사가 작성한 진단서와 유사하게 진단명, 관찰된 시각적 증상, 진단 이유를 포함하는 구조화된 텍스트를 구성했습니다. 모델이 단순 분류를 넘어 질환의 근거까지 설명할 수 있는지 확인하고자 했습니다.

Text2 데이터 샘플

진단명 : 유루증

사진에서 관찰된 증상 :

- 눈물 자국 및 지속적 눈물 흐름 (Epiphora) - 눈 내측 주변 털이 젖어 있으며...
- 눈물의 저류와 눈가 착색 (Tear staining) - 눈 아래 털에 눈물 자국이 관찰되며...

해당 증상에 따른 진단 이유 : 관찰된 증상들은 유루증의 시각적 기준과 일치하며...

2) 학습 결과

학습 지표(정량 평가) :

두 데이터셋으로 파인튜닝(Fine-tuning)된 모델의 성능을 비교한 결과, 예상과 달리

단순한 정보를 담은 Text1의 정량적 지표가 더 우수하게 나타났습니다.

- **Text1 (단순형)** : 전체 정확도 0.80, Macro Avg F1-score 0.81 을 기록하며 준수한 분류 성능을 보였습니다. 특히 유루증(F1: 0.90)과 무증상(F1: 0.85) 클래스에서 높은 성능을 보였습니다.

	precision	recall	F1-score	support
결막염	0.80	0.72	0.76	100
백내장	0.69	0.82	0.75	100
유루증	0.88	0.93	0.90	56
무증상	0.97	0.75	0.85	52
accuracy			0.80	308
macro avg	0.84	0.80	0.81	308
weighted vg	0.81	0.80	0.80	308

- **Text2 (상세형)** : 전체 정확도 0.72, Macro Avg F1-score 0.75 로 Text1 대비 전반적인 성능 하락이 관찰되었습니다. 특히 결막염(F1: 0.76 → 0.65)과 유루증(F1: 0.90 → 0.81)에서의 성능 저하가 두드러졌습니다.

	precision	recall	F1-score	support
결막염	0.68	0.63	0.65	100
백내장	0.67	0.71	0.69	100
유루증	0.79	0.82	0.81	56
무증상	0.84	0.83	0.83	52
accuracy			0.72	308
macro avg	0.75		0.75	308
weighted vg	0.72		0.72	308

실제 모델의 출력 결과(정성 평가) :

실제 테스트 이미지에 대한 모델의 추론 결과를 비교 분석하였습니다. 두 실험군 모

두 진단명은 도출하였으나, 생성된 텍스트의 스타일과 오류 양상에서 뚜렷한 차이를 보였습니다.

- [Text1 단순 진단형 모델 출력]

Text1 모델은 질환을 맞추는 데에는 비교적 성능이 높았지만, 이미지에 없는 일 반적인 의학 지식을 나열하거나(Hallucination), 문맥에 맞지 않는 문장으로 끝맺는 경향이 있었습니다.

Text1 예시 1 (결막염 케이스)

실제 진단명	결막염
예측 진단명	결막염
모델 응답	이 강아지는 결막염에 걸렸습니다.보이지 않는 증상 중 하나로, 결막 염은 강아지의 결막에 염증을 유발시킵니다.이는 일반적으로 자외선 노출, 먼지, 또는 바이러스에 의해 유발될 수 있습니다.증상으로는 눈이 붉고, 부종되고, 가려지는 것이 보일 수 있습니다.시각적으로 확인된 증상은 결막염의 유무를 판단하는 데 기여하지만, 확정적인 진단은 의학적 검사와 검사 결과에 기반한 시각적 판단만을 기술하십시오.
분석	정답은 맞쳤으나, 사진에서 확인할 수 없는 원인(자외선, 바이러스 등)을 백과사전식으로 나열하고, 학습 데이터에 포함된 투박한 어조 ("기술하십시오" 등)를 사용함.

Text1 예시 2 (백내장 케이스)

실제 진단명	백내장
예측 진단명	백내장
모델 응답	이 강아지는 백내장에 걸렸습니다.백내장은 안구의 유리체가 구리처럼 질병되어 투명하지 않게 되는 질병입니다.유리체 질병은 시각을 방해하고, 심하면 시각을 잃게 하기 때문에 즉시 치료가 필요합니다. 의학적 처리는 유리체를 제거하고, 필요에 따라 교정 유리체를삽입

	하는 수술을 포함합니다. 수술 후에는 시각 회복 기간이 필요하며, 회복기간 동안은 시각 보조구를 착용해야 합니다. 당국자들은 수...
분석	사용자에게 맞지 않는 과도한 치료 정보(수술, 보조구) 텍스트를 생성함. 또한 문장 생성 종료 시점을 찾지 못하고 텍스트를 계속 생성하는 문제가 관찰됨.

Text1 예시 3 (백내장 케이스)

실제 진단명	백내장
예측 진단명	백내장
모델 응답	이 강아지는 백내장에 걸렸습니다. 백내장은 유리체가混浊하는 질병으로, 시각을 방해하거나 시각을 잃게 하기로 알려져 있습니다. 시각 분석 결과에 기반한 시각적 판단으로, 백내장이 확진되었습니다.
분석	텍스트 생성 중 혼탁이라는 단어를 중국어, 또는 한자(混浊)로 표현하는 문제가 확인됨. 또한 어떤 시각적 증상이 관찰되었는지 설명하지 못함.

- [Text 2: 상세 서술형 모델 출력]

Text2 모델은 전문적인 진단 포맷을 잘 따랐고, 텍스트 생성 능력(한국어), 시각적 증상 설명 능력이 향상되었습니다. 그러나 EOS 토큰 문제로 인해 생성을 멈추지 못하고 마지막 문장을 반복 생성하는 현상이 Text1보다 빈번하고 심각하게 발생했습니다.

Text2 예시 1 (백내장 케이스)

실제 진단명	백내장
예측 진단명	백내장
모델 응답	진단명: 백내장/초기- 사진에서 관찰된 증상:1. 국소적 흐림: 렌즈의 중심부 또는 후부 피질 영역에 작고 국한된 흐림이나타납니다. 사진에서 렌즈 내에 작은 흐림이 관찰됩니다.2. 안저 반사 유지: tapetal reflex 가 완전히 유지되며 반사광이 눈에 보입니다. 사진에서 안저 반사가 유지되고 있습니다.3. 공포 형성: 투명한 공기방을 모양의

	vacuole 이 렌즈 내에서 관찰...
분석	'국소적 흐림', '안저 반사', '공포 형성' 등 전문적인 시각적 특징을 매우 구체적으로 서술함. 그러나 EOS 토큰 문제로 모델이 텍스트 생성을 멈추지 못하고 Max Token이 가득 찰 때까지 마지막 문장을 계속 반복함.

Text2 예시 2 (유루증 케이스)

실제 진단명	유루증
예측 진단명	유루증
모델 응답	진단명: 유루증- 사진에서 관찰된 증상:- 눈물 자국 및 지속적 눈물 흐름 (Epiphora)- 눈 내측 주변 털이 젖어 있으며, 눈물로 인해 털이 축축하게 젖어 있는상태가 관찰됩니다.- 눈물의 저류와 눈가 착색 (Tear staining)- 눈 밑 털에 눈물 자국이 명확히 보이며, 착색이 관찰됩니다.- 해당 증상에 따른 진단 이유: 관찰된 증상들은 유...
분석	정Text1 실험에 비해 구조화된 응답을 해주고, 중국어를 쓰거나 이상한 말을 하는 등 텍스트 생성 능력 문제가 사라짐.

3) 문제점 및 분석

정성 및 정량 평가를 종합해 볼 때, 1차 실험에서 드러난 주요 문제점은 다음과 같습니다. 이는 다음 단계 실험의 개선 방향성을 설정하는 중요한 근거가 되었습니다.

- **종료 토큰(EOT Token) 부재로 인한 반복 생성 오류 :** 학습 데이터 전처리 과정에서 문장의 끝을 알리는 EOS 토큰(<|endfotext|>)이 명시적으로 포함되지 않는 실수가 있었습니다. 이로 인해 모델은 문장 생성이 완료되었음에도 불구하고 멈추지 않고, max_token 한도에 도달할 때까지 마지막 문장을 무의미하게 반복 생성하는 현상이 발생했습니다. 이는 불필요한 추론 시간 증가와 후처리(Post-processing)의 어려움을 초래했습니다.
- **과업 복잡도 증가와 라벨 노이즈 (Label Noise) :** Text2 의 상세한 설명 문구들(예: "관찰됩니다", "보입니다" 등)과 긴 문장 구조가 모델에게는 학습해야

할 패턴의 복잡도를 급격히 높였습니다. 모델이 핵심 질환 특징(Feature)보다는 문제나 부가적인 단어들을 학습하는 데 자원을 소모한 것으로 판단됩니다.

- **공통 키워드로 인한 클래스 간 혼동** : '결막염'과 '유루증'의 상세 설명에 '눈물', '분비물', '충혈'과 같은 유사한 단어들이 공통적으로 포함되면서, 모델이 이 두 질환을 명확히 구분하는 데 어려움을 겪었습니다. 텍스트가 길어질수록 클래스 간의 경계가 모호해지는 부작용이 발생했습니다.
- **데이터의 단편화** : Text1과 Text2로 데이터셋을 이분화하여 실험함으로써, 전체 데이터의 잠재력을 온전히 활용하지 못했습니다. 또한 단순한 텍스트 포맷의 차이만으로는 모델의 진단 정확도를 획기적으로 높이는 데 한계가 있음을 확인했습니다.

이러한 결과를 바탕으로 다음 실험에서는 EOS 토큰 문제를 해결하고 텍스트의 형식을 다양화하여 과적합과 반복 생성을 방지하며, 텍스트 정보에만 의존하는 한계를 극복하기 위해 이미지 처리 방식(RoI)을 개선하는 방향으로 전략을 수정하였습니다.

5.2 중간 실험 : 템플릿 다양화 (Diagnosis & Chatbot)

이전 4.1.1 실험(Text2)에서 상세한 서술형 데이터가 오히려 모델의 성능을 저하시키고, 특정 문제에 과적합(Overfitting)되는 문제를 확인되었습니다. 이를 해결하기 위해 이번 실험에서는 데이터 템플릿을 다양화하여 텍스트의 단편화를 방지하고, 모델이 다양한 질의응답 형태에 적응할 수 있도록 챗봇 데이터를 추가하여 학습을 진행했습니다. 또한 텍스트 데이터에 EOS 토큰이 없어도 학습 파이프라인에서 이미지와 텍스트에 자동으로 필요한 토큰을 적용하도록 구성했습니다.

1) 데이터셋 구성

학습 데이터의 텍스트 포맷을 다변화하여, 모델이 단순한 진단문 생성뿐만 아니라 사용자와의 상호작용 상황까지 학습하도록 설계했습니다.

- **텍스트 템플릿 다양화 (Diagnosis & Chatbot)** : 기준의 고정된 진단서 포맷(Diagnosis)에 더해, OpenAI API 를 활용하여 생성한 챗봇 데이터를 추가했습니다. 하나의 질환 이미지에 대해 사전에 작성한 6 가지 챗봇 템플릿 종류 중 3 가지를 랜덤으로 선택하고 예상 질문과 답변 쌍(Q&A)을 구축하여, 다양한 질문 의도와 어조에 대응하도록 했습니다.
- **데이터 구성 샘플**
 - **Diagnosis (진단형)** : 공식 소견서, 수의사 브리핑 포맷 등 마크다운으로 구조화된 텍스트
 - **Chatbot (대화형)** : 구어체 질문과 이에 대한 응답.
 - **데이터 예시** : 하나의 이미지에 대해 1 개의 진단형 데이터와 3 개의 챗봇형 데이터로 구성했습니다.

진단형 데이터

```
"instruction": "공식 소견서 형식으로 작성해줘. 마크다운을 사용하고, '# 진단명',  
'## 증상' 헤더를 포함해줘.", "answer": "# 진단명\n결막염\n## 증상\n- 결막  
충혈 (Conjunctival hyperaemia)\n- 눈물 과다 및 분비물 (Lacrimation &  
Discharge)", "token": "report"
```

챗봇형 데이터 1

```
"instruction": "이 강아지 눈 사진에 대한 진단명과 모든 증상을 알려줘.", "answer":  
"진단명은 결막염입니다. 관찰된 증상은 결막 충혈 (Conjunctival hyperaemia)과 눈  
물 과다 및 분비물 (Lacrimation & Discharge)입니다.", "token": "chatbot"
```

챗봇형 데이터 2

```
"instruction": "수의사처럼 보호자에게 진단/증상을 설명해줘.", "answer": "안녕하세  
요. 강아지의 눈 상태를 확인해본 결과, 결막염으로 진단됩니다. 결막 충혈과 함께
```

눈물 과다 및 분비물이 나타나고 있습니다. 이 증상은 불편함을 줄 수 있으니, 적절한 치료를 권장합니다.", "token": "chatbot"

챗봇형 데이터 3

"instruction": "우리 강아지 상태가 괜찮은지 설명해줘(권고 포함).", "answer": "강아지의 상태가 좋지 않은 것 같아요. 결막염으로 인해 결막 충혈과 눈물 과다 및 분비물이 관찰됩니다. 수의사와 상담하여 적절한 치료를 받는 것이 좋습니다.", "token": "chatbot"

2) 학습 결과

- 학습 지표(정량 평가)**: 새로운 데이터 구성을 적용한 결과, 모델의 전체 정확도는 0.7086, F1-Score는 0.6935를 기록했습니다.

Class	Precision	Recall	F1-score	Support
무증상	0.69	0.95	0.80	100
결막염	0.76	0.82	0.79	100
궤양성각막질환	0.92	0.83	0.87	100
백내장	0.79	0.80	0.79	100
안검내반증	0.50	0.52	0.50	100
유루증	0.57	0.26	0.35	100
Accuracy			0.7086	

- 모델의 출력 결과(정성 평가)**: 1 차 실험(Text2)의 고질적인 문제였던 반복 생성 오류는 해결되었으며, 진단 형식의 안정성은 확보되었습니다. 그러나 챗봇 기능의 유연성은 기대에 미치지 못했습니다
 - Diagnosis (진단 소견서)**: 학습된 템플릿에 맞춰 진단명과 상세 증상을 구조적으로 잘 서술하였습니다.

진단 소견서 예시 1 (백내장 케이스)

실제 진단명	백내장
예측 진단명	백내장
모델 응답	<p># 진단명 백내장/초기</p> <p>## 증상</p> <ul style="list-style-type: none"> - 국소적 흐림: 렌즈의 중심부 또는 후부 피질 영역에 작고 국한된 흐림이 나타납니다. - 안저 반사 유지: tapetal reflex가 완전히 유지되며 반사광이 눈에 보입니다. - 염증 없음: 이 단계에서는 안구 내 염증이나 통증 소견은 관찰되지 않습니다.
분석	전문 용어를 사용하여 증상을 구체적으로 묘사하였으며, 문장 잘림 없이 완결된 형태를 보임.

- **Chatbot (대화형 응답) :** 다양한 질문에 대응하기 위해 챗봇 데이터를 학습 시켰음에도, 모델은 질문의 구체적인 내용과 관계없이 학습 데이터에 존재하는 고정된 답변 패턴만을 반복했습니다.

챗봇 예시 2 (챗봇 형식 질의응답 실패 사례)

질문	백내장 수술 비용은 대략 얼마 정도 해? 그리고 수술 안 시키고 집에서 관리하는 방법은 없어? → (외부 지식 및 구체적 조언 요구)
모델 응답	강아지의 눈 상태는 백내장으로 보입니다. 관찰된 증상은 수정체의 혼탁입니다. 가까운 병원에 내원하여 진료를 받으시길 권장합니다.
분석	사용자가 질문한 '수술 비용'이나 '집에서 관리 방법'에 대한 답변은 전혀 하지 못함. 엉뚱하거나 구체적인 질문에도 학습 데이터셋의 평균적인 답변 길이와 형식(진단 + 내원 권유)에 갇힌 경직된 답변을

	출력함.
--	------

3) 문제점 및 분석

이번 실험을 통해 텍스트 단편화 문제는 일부 개선되었으나, '지능형 진단'의 한계와 '미세 병변 탐지 능력 저하'라는 새로운 과제가 도출되었습니다.

- **챗봇 응답의 경직성(Rigidity) 및 RAG 적용 실패**

질문의 의도를 파악하여 유동적으로 대처하기보다, 학습된 답변 템플릿을 그대로 출력하는 경향이 강했습니다. 특히, 진단 외의 정보(비용, 관리법 등)를 제공하기 위해 외부 정보(Web Search, RAG)를 활용하도록 유도하려 했으나, 모델은 외부 정보를 가져와 답변에 반영하는 방법을 학습하지 못하고, 단순히 학습된 데이터의 형식만을 모방했습니다.

- **시각적 정보 집중력 저하**

텍스트 생성 태스크가 다양해지고 복잡해지면서, 모델이 이미지의 핵심 병변에 집중하는 능력이 상대적으로 약화되었습니다. 특히 유루증(F1 0.35)과 같이 눈 자체가 아닌 '눈 밑 털의 착색' 등 주변부 특징을 봐야 하는 질환에서 성능이 많이 낮은 것으로 확인되었습니다. 이는 모델이 전체 이미지에서 질환과 관련된 관심 영역(ROI)을 스스로 찾는데 어려움을 겪고 있는 것으로 추정됩니다.

- **새로운 진단 라벨 추가**

1 차 실험에서는 신속한 개발 환경 구축과 모델의 기본 성능 검증을 위해 결막염, 백내장, 유루증, 무증상의 4 개 핵심 라벨만을 사용하여 학습을 진행하였습니다. 이후 2 차 실험에서는 실제 진단 환경을 반영하기 위해 총 7 가지 전체 라벨로 범위를 확장하여 학습을 수행하였습니다. 그러나 라벨이 세분화됨에 따라 안검염, 유루증, 안검내반증 3 가지 병변에서 진단 정확도가 현저히 저하되는 현상이 발생하였습니다. 이는 해당 질환들이 눈 주변의 발적,

부종, 눈물 자국 등 시각적 특징을 공유하고 있어 모델이 이를 명확히 구분하는데 한계가 있었기 때문입니다. 아래 표의 혼동 행렬(Confusion Matrix)을 통해 실험 결과를 상세히 분석해보면, 유사 병변 간의 오분류 경향이 뚜렷하게 드러납니다. 특히 유루증의 경우, 실제 정답을 맞춘 건수는 26 건에 불과한 반면, 안검내반증으로 오분류된 건수는 36 건으로 정답보다 오답의 빈도가 더 높게 나타났으며, 안검염으로 예측한 경우도 19 건이나 발생했습니다. 이는 모델이 유루증의 특징을 안검내반증이나 안검염과 혼동하고 있음을 나타냅니다.

마찬가지로 안검내반증 또한 정답(52 건) 외에 안검염(18 건)과 유루증(16 건)으로 분산되어 예측되는 등, 전반적으로 이 세 가지 클래스 간의 경계가 모호하게 학습되었음이 수치적으로 확인되었습니다. 결과적으로 시각적으로 매우 유사한 증상을 가진 질환들에 대해서는 단순한 라벨 확장만으로는 높은 정확도를 확보하기 어려우며, 이를 해결하기 위한 라벨 병합이나 추가적인 데이터 전처리가 필요함을 알게 되었습니다.

	무증상	결막염	궤양성각막질환	백내장	안검내반증	안검염	유루증
무증상	95	0	3	2	0	0	0
결막염	3	82	2	13	0	0	0
궤양성각 막질환	8	5	83	4	0	0	0
백내장	3	14	1	80	0	2	0
안검내반 증	11	2	0	1	52	18	16
안검염	1	2	0	0	16	78	3
유루증	15	2	1	1	36	19	26

위와 같은 분석 결과를 토대로, 최종 실험에서는 진단 본연의 정확도를 극대화하는데 주안점을 두고 실험 계획을 수립했습니다. 우선, 미세한 병변의 식별력을 높이기 위해 YOLO 모델을 활용한 ROI(Region of Interest) 기반의 이미지 크롭 전처리

과정을 도입하여 모델이 시각 정보에 온전히 집중할 수 있도록 개선하였습니다. 이는 챗봇의 대화 유연성을 확보하기보다, 의료 보조 도구로서의 신뢰성을 확보하는 것이 우선이라는 판단에 따른 것입니다.

또한 단순 진단을 넘어 상세한 결과 리포트를 제공하기 위해 데이터셋을 새롭게 구축하였습니다. 새롭게 구축될 데이터셋은 진단서 하단의 상세 소견 작성과 더불어, 모델이 웹 검색(Web Search) 및 검색 증강 생성(RAG) 기술을 활용하여 근거 기반의 답변을 생성하는 방법을 학습하는데 초점을 맞출 것입니다.

앞선 실험들의 결과와 분석 내용을 바탕으로, 본 프로젝트의 최종 목표인 정확한 진단과 풍부한 설명을 모두 달성하기 위한 마지막 실험을 진행하였습니다. 이 단계에서는 모델의 입력 방식을 기본적으로 개선하고, 구축된 모든 유형의 데이터셋을 통합하여 학습시켰습니다.

5.3 최종 실험 : 데이터 통합 및 ROI 최적화

1) 실험 설계 및 전략 수립

최종 실험에 앞서, 중간 단계로 Multi-Image-Input(원본 이미지 + 크롭 이미지) 전략 도입과 새로운 데이터셋 사용을 시도했습니다. 그러나 원본 이미지와 안구 영역(ROI) 크롭 이미지를 동시에 입력하여 상호 보완적인 정보를 학습시키려던 의도와 달리, 오히려 단일 이미지를 사용할 때보다 진단 정확도가 하락하는 현상이 관찰되었습니다. 실패 원인을 분석한 결과, 모델이 두 이미지 간의 특징 정보를 유기적으로 연결하지 못했을 뿐만 아니라, 단순 크롭(Simple Crop) 과정에서 병변 판단에 필수적인 주변부 문맥 정보가 소실되었기 때문으로 파악되었습니다.

또한, 데이터 측면에서는 기존의 '진단명+증상' 데이터를 마크다운에서 JSON 형식으로 구조화하고, OpenAI API를 활용해 진단 설명용 텍스트를 새롭게 생성하여 학습에 활용하였습니다. 하지만 확보된 데이터셋의 규모가 진단 데이터 5,600개, 설명 데이터 1,020개(단순 3배 증강 적용하여 3,060개)에 불과하여, 모델이 복잡한 질환 패턴과 자연스러운 설명을 동시에 학습하기에는 절대적인 데이터 양이 부족했던 것으로 판단됩니다.

결론적으로 이 중간 실험을 통해 단순한 입력 정보의 나열이나 소규모 데이터 증강만으로는 성능 개선에 한계가 있음을 확인하였으며, 이는 최종 실험에서 ROI 추출 기법 개선과 대규모 데이터 확보의 필요성을 보여줬습니다.

이에 따라 최종 실험에서는 다음과 같은 개선된 전략을 수립하였습니다.

- **ROI 전략 수정 (Padding Crop)**

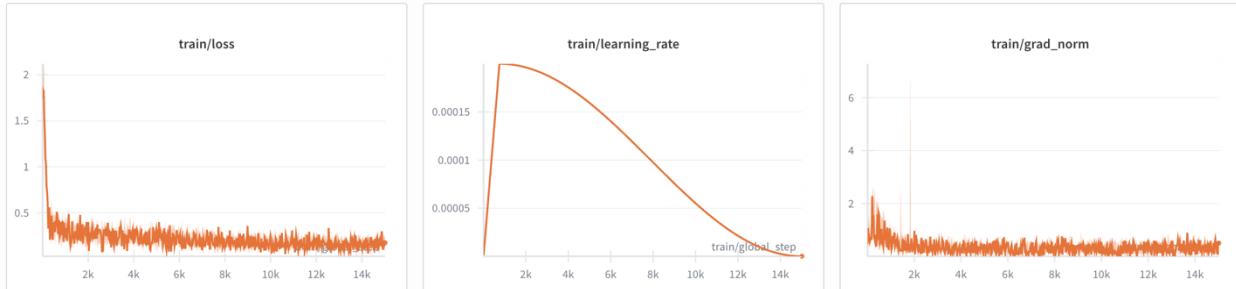
두 장의 이미지를 넣는 대신, YOLO 모델로 탐지한 좌표에 적절한 패딩을 주어 눈 주변부(털, 피부 등)까지 포함한 단일 크롭 이미지를 사용했습니다. 이는 모델이 병변과 주변 문맥을 한 번에 파악하도록 도와줍니다.

- **데이터셋 통합**

진단의 정확성과 응답의 유연성을 모두 확보하기 위해 가용 가능한 모든 형태의 데이터셋을 학습에 투입했습니다.

- **진단형** : 마크다운 형식 5,600 개 + JSON 형식 5,600 개
- **진단 설명형** : 질병에 대한 상세 백과사전식 설명 데이터 1,020 개 (단순 2 배 증강하여 2,040 개)
- **챗봇형** : 이전 실험에서 활용했던 대화형 데이터 (5,600 개 * 3 가지 템플릿으로 총 16,800 개)

2) 최종 학습 개요



- **학습 환경 (Hardware)** : NVIDIA A100 80GB PCIe (1 GPU)
- **소프트웨어 버전** : Python 3.12.3 / Linux-6.8.0
- **총 학습 시간** : 10 시간 59 분 13 초
- **총 학습 단계 (Steps)** : 15,020 Steps
- **최종 학습 손실 (Loss)** : 약 0.2 수준 수렴

3) 학습 결과

• 학습 지표(정량 평가)

최종 전략을 적용한 결과, 전체 정확도는 0.74, F1-Score는 0.7404를 기록하며 모든 실험 중 가장 우수한 성능을 달성했습니다.

- **성능 향상** : 2 차 실험에서 0.70 대였던 정확도가 0.74로 약 4%p 상승했습니다.
- **유루증 성능 향상** : 단순 크롭 시 0.35 까지 떨어졌던 유루증의 F1-Score가 0.4772로 향상되었습니다. 이는 패딩을 통해 눈 밑 털의 착색 정보를 보존한 전략이 유효했음을 보여줍니다. 하지만 여전히 안검내반증이나 안검염과 혼동되는 경향이 있어, 시각적으로 유사한 질환 간의 구분은 여전히 어렵다는 것을 알 수 있었습니다.

Class	Precision	Recall	F1-score	Support
무증상	0.91	0.93	0.92	100
결막염	0.80	0.82	0.81	100
궤양성각막질환	0.93	0.87	0.90	100
백내장	0.87	0.86	0.86	100
안검내반증	0.53	0.71	0.60	100
안검염	0.74	0.52	0.61	100
유루증	0.48	0.47	0.48	100
Accuracy			0.7400	

- 실제 모델의 출력 결과(정성 평가)

최종 모델은 사용자의 요청 의도에 따라 완벽하게 다른 포맷의 답변을 생성하는 능력을 갖추었습니다.

Case 1 : 시스템 연동을 위한 구조화된 진단 (JSON)

애플리케이션 등에서 후처리가 용이하며, 텍스트 생성 속도가 빠르다는 장점이 있습니다.

모델 출력 예시
<pre>{ "diagnosis": "궤양성각막질환", "symptoms": ["각막 표면의 혼탁 또는 회백색 탁음", "각막 상처 또는 함몰된 부위", "삼출물과 눈곱 형성"] }</pre>

Case 2 : RAG 기반 상세 설명

단순 챗봇의 경직된 답변 대신, 모델이 프롬프트에 제공된 외부 지식(Web Search Context)을 활용하여 전문적이고 근거 있는 설명을 생성합니다.

모델 출력 예시 - 유루증 설명

질병에 대한 설명

유루증은 강아지의 눈에서 눈물이 과다하게 생성되거나, 눈물 배출이 원활하지 않아 눈가가 지속적으로 젖는 상태를 말해요. 이로 인해 눈가에 눈물 자국이 남고, 눈가가 갈색이나 붉은색으로 변색될 수 있어요 [2]. 이러한 상태는 강아지에게 경미한 피부 자극을 유발할 수 있어요 [2].

진단 근거

유루증을 진단할 때는 눈가의 지속적인 젖음과 변색을 관찰하는 것이 중요해요. 또한, 결막이 충혈되었는지, 안검염이 동반되었는지를 확인해야 해요 [2]. 눈물의 흐름이 원활하지 않거나 눈물점이 막혀 있을 때, 눈물점 탐침이나 식염수 세정을 통해 개통 여부를 평가하기도 해요 [2].

주요 발생 원인유루증의 원인은 여러 가지가 있어요. 코눈물관의 협착이나 폐쇄, 또는 안검이나 속눈썹의 이상이 원인이 될 수 있어요 [2]. 또, 자극이나 알레르기, 그리고 눈꺼풀의 형태 문제도 유루증을 유발할 수 있어요 [2]. 이 외에도 이물질에 의한 폐색이나 염증 후 협착도 원인이 될 수 있어요 [2].

관리 방법

가정에서 관리할 때는 미온 생리식염수와 멸균 거즈를 사용하여 눈가를 부드럽게 닦아주는 것이 중요해요 [2]. 눈가의 위생을 잘 유지하고 젖은 부분은 건조하게 관리해야 해요 [2]. 먼지나 바람, 샴푸 등의 자극원을 최소화하는 것도 중요하고, 안검이나 속눈썹 이상이 의심될 경우 수의사와 상담을 통해 교정 가능성을 검토해야 해요 [2]. 피부 염증이 동반되거나, 분비물의 성상이 변화할 경우에는 전문적인 평가가 필요해요 [2].

4) 결론 및 의의

최종 실험을 통해 본 연구는 단순한 질병 분류를 넘어, 임상적 효용성을 갖춘 진단 보조 시스템으로서의 가능성을 확인했습니다.

- **객관적 성능 검증 및 경쟁력 확보**

객관적인 성능 평가를 위해 대표적인 이미지 분류 모델인 ResNet50(Accuracy 0.79)과 멀티모달 모델인 CLIP(Accuracy 0.73)을 베이스라인으로 선정하여 비교 실험을 진행했습니다. 실험 결과, 본 모델은 이들 베이스라인 모델과 비교했을 때 대등하거나 우수한 분류 성능을 보였으며, 단순 레이블 예측(Classification)을 넘어 질환의 원인과 양상을 텍스트로 설명(Generation)할 수 있다는 점에서 차별화된 경쟁력을 입증했습니다.

- **ROI 최적화와 문맥 인식**

무조건적인 크롭보다는 주변 문맥을 살리는 Padding Crop 이 진단 성능, 특히 주변부 병변 탐지에 필수적임을 확인했습니다. 이는 병변 자체뿐만 아니라 주변 피부와 털의 상태까지 종합적으로 고려해야 하는 안과 질환의 특성을 반영합니다.

- **유사 병변 그룹 통합 시 높은 신뢰도 (Accuracy 0.92)**

실험 초기, 시각적 특징(눈 주변 염증, 부종, 눈물 자국)이 매우 유사한 유루증, 안검내반증, 안검염 세 가지 클래스 간에 오분류(Confusion)가 빈번하게 관찰되었습니다. 그러나 이를 임상적 유사 그룹(Periocular Issues Group)으로 묶어서 재분석한 결과, 우리 모델의 진단 가치는 더욱 명확해졌습니다.

[통합 레이블 기준 정확도 비교]

- **CLIP (Baseline 1)** : 0.897 (약 90%)
- **ResNet50 (Baseline 2)** : 0.944 (약 94%)
- **Ours (제안 모델)** : 0.920 (약 92%)

본 모델의 통합 정확도(0.92)는 동일한 멀티모달 아키텍처인 CLIP(0.90)을 상회하는 수치이며, 순수 이미지 분류에 특화된 ResNet50(0.94)과 비교해서도 매우 근접한 성능입니다. 이는 우리 모델이 "이 강아지의 눈 주변에 문제가 있다"는 사실을 포착하는 1차 스크리닝 능력 면에서 전문 분류 모델에 뒤지지 않는 높은 신뢰도를 가짐을 나타냅니다. 실제로 모델 예측 사례를 분석해보면, 유루증 예측(81건) 중 100%가 유사 병변 그룹 내 정답, 안검내반증 예측(86건) 중 100%가 유사 병변 그룹 내 정답을 기록하며, 보호자가 놓칠 수 있는 안구 주변의 이상 징후를 조기에 포착하고 경고하는 데 있어 강력한 효용성을 증명했습니다.

- **확장성 및 향후 과제**

본 모델은 단일 모델로 JSON, Markdown, 구어체 등 다양한 포맷 대응이 가능하며, RAG 기술과 결합하여 최신 의학 정보를 반영할 수 있는 기반을 마련했습니다. 향후 양질의 이미지 데이터와 임상 정보(견종, 나이 등)를 추가로 결합한다면, 유사 그룹 내 세부 질환 분류의 정확도 또한 ResNet50 수준 이상으로 개선될 것으로 기대됩니다.

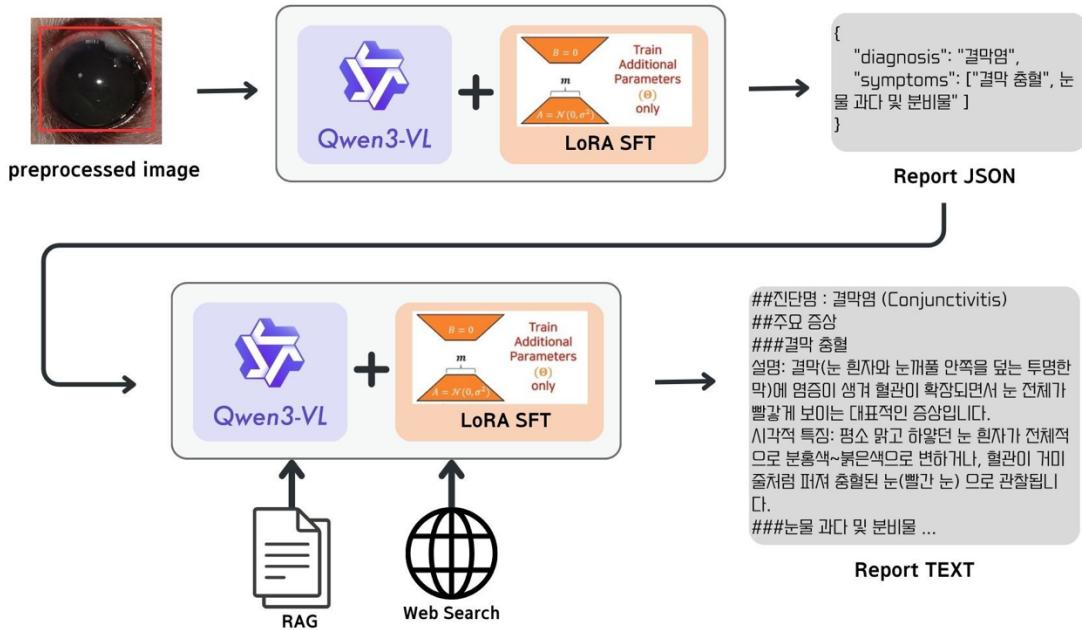
6. 아키텍쳐

6.1 최종 모델 아키텍쳐

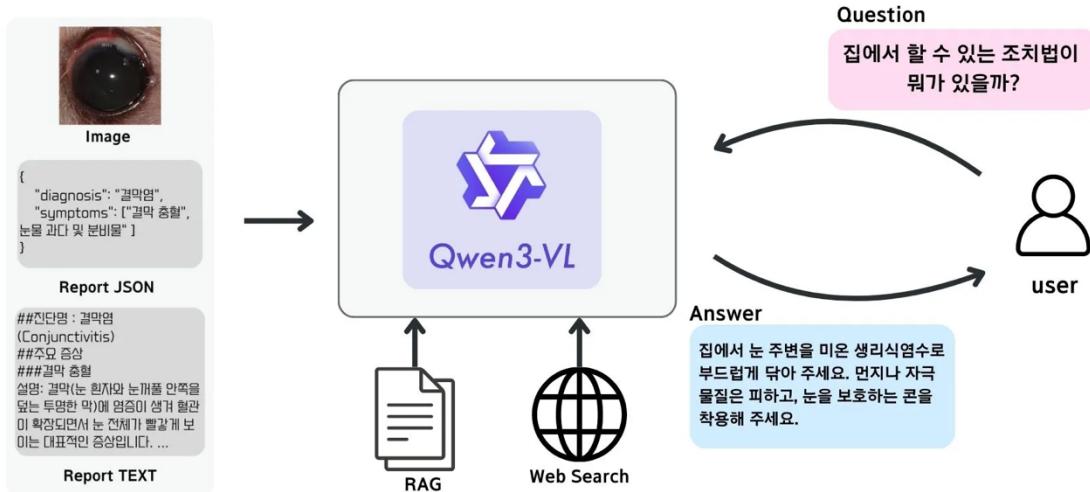
1. Image Pre-Processing



2. REPORT



3. CHATBOT



본 시스템은 반려동물 안구 이미지를 기반으로 질병을 분류하고, 진단 결과를 구조화된 형태(JSON)와 서술형 보고서(TEXT)로 생성하며, 사용자의 추가 질문에 대응하는 챗봇 기능을 통합한 End-to-End Vision-Language Diagnosis Pipeline이다. 핵심 구성 요소는 YOLOv8 기반 이미지 전처리, Qwen3-VL + LoRA 기반 보고서 생성 모듈, 그리고 RAG·Web Search를 활용한 대화형 질의응답 모듈로 구성된다.

전체 파이프라인은 다음과 같이 3단계로 구성된다.

1) 이미지 전처리 (Image Pre-Processing)

사용자가 업로드한 원본 이미지는 조명·배경·촬영 각도 등 다양한 품질 변동성을 가지므로, 진단에 필요한 안구 영역만을 정확히 추출하는 과정이 필요하다.

- YOLOv8 Object Detection 을 사용해 안구 및 병변 위치를 자동 검출
- Bounding Box 기반 cropping 을 수행하여 preprocessed image 생성
- 이후 단계에서 모델 입력 품질을 보장하고 오진 가능성 감소

결과: 진단 모델이 분석하기 적합한 형태의 표준화된 이미지

2) 보고서 생성 (REPORT)

보고서는 두 개의 서브 프로세스로 구성된다.

1. JSON 형태의 구조화 진단 생성

- Qwen3-VL 에 LoRA SFT 적용
- 질환명, 핵심 증상 등을 JSON 구조로 생성

```
{  
    "diagnosis": "결막염",  
    "symptoms": ["결막 충혈", "눈물 과다 및 분비물"]  
}
```

이 단계는 후속 텍스트 생성, 챗봇 대화 작업의 기반 데이터로 활용된다.

3) 서술형 진단 보고서 생성

1단계에서 생성된 JSON, RAG 문서, Web 검색 정보를 통합한 뒤, Qwen3-VL + LoRA 모델을 다시 호출하여 설명형 보고서를 생성한다.

- 질병에 대한 설명
- 진단 근거
- 주요 발생 원인
- 관리 방법
- 사용자의 이해를 돋는 자연어 기반 상세 설명

3) 챗봇 모듈 (Chatbot)

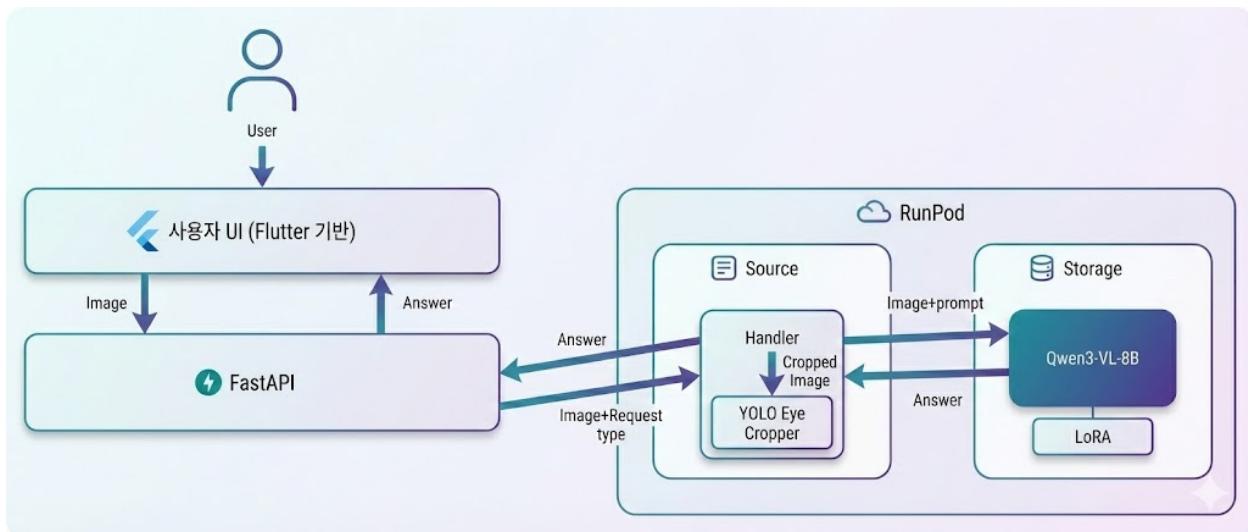
사용자가 질문(예: “집에서의 관리 방법은?”)을 입력하면 다음 정보 기반으로 답변을 생성한다.

- 이미지 정보
- 진단 JSON
- 상세 Report TEXT
- RAG + Web Search

이를 통해 개별 환자 상황에 맞춘 대화형 의료 가이드 제공이 가능하다.

6.2 전체 시스템 아키텍쳐

6.2.1 시스템 구성 개요



6.2.2 모듈별 아키텍쳐 상세 설명

본 프로젝트의 시스템 아키텍처는 크게 플러터 기반의 프론트 엔드, FastAPI 기반의 백엔드, RunPod Serverless 환경 3 가지로 나누어 볼 수 있다.

1) 프론트엔드 (Flutter 기반 사용자 UI)

Flutter로 구현된 사용자 앱은 사진 업로드, 진단 요청 버튼, 결과 화면 구성 등 사용자 인터페이스 전반을 담당한다. 사용자는 앱 내에서 반려견의 안구 사진을 업로드하며, 해당 이미지는 FastAPI 백엔드로 전달된다.

주요 기능 :

- 이미지 업로드 및 미리보기
- FastAPI로 진단 요청 전송
- 분석 결과(질환명, 설명)를 사용자에게 표시

2) FastAPI 백엔드

FastAPI 백엔드는 프론트엔드와 RunPod 서비스 함수 사이의 중계 및 처리 로직을 담당한다. 프론트엔드에서 전달된 이미지와 요청 모드를 수신하여 RunPod 서비스 엔드포인트로 전달한다. 또한 RunPod에서 반환된 모델의 추론 결과를 가공하여 프론트엔드에서 사용할 수 있도록 변환한다.

주요 역할:

- 이미지 및 요청 파라미터 검증
- RunPod Serverless API 호출
- 모델 응답(JSON/텍스트) 정제 후 Flutter로 반환
- 상태 코드 및 예외 처리

3) RunPod Serverless 환경

RunPod는 본 프로젝트에서 모델 추론의 핵심 컴퓨팅(Core Computing) 역할을 담당하는 클라우드 플랫폼으로 서비스(Serverless) 방식을 채택하여 인프라 관리 소요를 줄이고 효율적인 리소스 운영을 가능하게 한다.

내부 구성 요소

Runpod 환경은 크게 실행 제어(Handler)와 모델 엔진(Model Storage)의 두 가지 요소로 구성된다.

A. Handler (Serverless Function)

GitHub 저장소와 연동된 `handler.py` 는 RunPod 서비스의 엔트리 포인트(Entry Point)이자, 실질적인 '**모델 호출 전달 API 서버**' 역할을 수행한다.

• 주요 워크플로우

1. **요청 수신** : FastAPI로부터 전달받은 데이터(이미지, 요청 모드 등) 파싱
2. **전처리** : 모델 입력 형식에 맞는 데이터 가공 수행

3. 모델 호출 : 로드된 Qwen3-VL-8B 모델에 데이터 전달 및 추론 실행
4. 응답 반환 : 생성된 진단 결과를 FastAPI로 반환

B. Inference Model (RunPod Storage)

추론을 수행하는 AI 모델은 RunPod Storage에 저장되어 있으며, Handler가 이를 직접 로드하여 사용한다.

- 모델 스펙
 - Base Model : Qwen3-VL-8B (Vision-Language Model)
 - Fine-tuning : 반려견 암구 질환 도메인에 특화된 LoRA 파인튜닝 적용
- 주요 특징
 - 멀티모달 입력 : 이미지와 프롬프트를 결합하여 구조화된 진단 결과 생성
 - 고속 추론 : Handler와 모델이 RunPod 내부 로컬 환경에서 통신하므로, 외부 네트워크 지연(Latency) 없이 빠른 속도로 추론 가능

6.2.3 전체 데이터 흐름 요약

본 아키텍처의 전체 동작 프로세스는 다음과 같은 순서로 진행됩니다.

1) 사용자 입력 (User Interaction)

- [User -> Flutter App]
- 사용자가 반려견의 암구 이미지를 업로드하고, 원하는 기능(진단 등)을 선택하여 요청을 시작함.

2) API 요청 전송 (Client to Backend)

- [Flutter -> FastAPI]
- 앱은 업로드된 이미지와 선택된 요청 모드(Diagnosis, Chatbot 등)를 패키징하여 메인 서버로 전송함.

3) 추론 엔드포인트 호출 (Backend to AI Cloud)

- [FastAPI -> RunPod Serverless]
- FastAPI 서버는 수신된 데이터를 기반으로 RunPod의 Serverless Endpoint를 호출하여 Image와 Mode 데이터를 전달함.

4) 모델 추론 수행 (Inference Execution)

- [Handler -> Qwen3-VL-8B Model]
- RunPod Handler는 입력된 이미지와 프롬프트를 전처리한 후, 메모리에 로드된 LoRA 파인튜닝 모델에 주입하여 추론을 수행함.

5) 결과 반환 (Response Pipeline)

- [Model -> Handler -> FastAPI]
- 모델이 생성한 진단명, 증상 설명 등의 텍스트 결과가 Handler를 거쳐 FastAPI 서버로 구조화되어 반환함.

6) 결과 시각화 (Result Rendering)

- [FastAPI -> Flutter]
- 최종적으로 앱은 UI에 진단 결과를 표시하여 사용자에게 분석 내용을 전달함.

6.2.4 설계 선택 이유와 장점

1) RunPod Serverless 사용

- GPU 인스턴스를 상시 유지할 필요 없이 요청 시에만 비용 발생
- GitHub 연동으로 코드 자동 배포 가능
- 대규모 VLM 모델을 안정적으로 운영할 수 있는 환경 제공

2) 모델 + LoRA 스토리지 분리

- 모델 재배포 없이 LoRA 가중치만 업데이트 가능
- 도메인 특화 성능 향상

3) FastAPI를 중간 API 레이어로 채택

- Flutter와 RunPod 간 복잡한 로직을 프론트가 직접 처리할 필요 없음
- 서비스와의 연결 관리, 로깅, 예외 처리 담당
- 향후 확장(로그 저장, 사용자 인증, 요약 API 추가 등)에 유리

4) 무상태(Stateless) 구조 → 데이터베이스 미사용

- 모델 진단 결과는 저장하지 않고 즉시 반환하는 구조
- 시스템 유지보수 및 배포의 단순화

7. 결론

7.1 기대 효과

본 연구에서 제안한 VLM 기반 반려동물 안구·피부 질환 진단 보조 시스템은 기술적·실용적 측면에서 다음과 같은 기대 효과를 가진다.

첫째, 임상적으로 활용 가능한 수준의 1차 스크리닝 도구로 기능할 수 있다. 제안 모델은 기존 CNN·CLIP 기반 분류기와 비교하여 동등 이상 또는 근접한 분류 성능을 보이면서도, 질병명뿐 아니라 핵심 증상과 시각적 근거를 함께 제시한다. 특히 안구 주변 염증·부종·눈물 자국 등을 하나의 “주변부 병변 그룹”으로 통합해 평가했을 때,

통합 정확도 0.92 수준을 달성함으로써, “눈 주변에 이상이 있는지 없는지”를 구분하는 1차 선별 단계에서 높은 신뢰도를 확보하였다. 이는 보호자가 질환의 존재 여부를 조기에 인지하고, 병원 내원 여부를 합리적으로 결정하는 데 실질적인 도움을 줄 수 있다.

둘째, 설명 가능한 AI(Explainable AI) 기반의 사용자 신뢰도 향상이라는 측면에서 의미가 있다. 본 시스템은 단순한 레이블 출력에 그치지 않고, ROI 기반 전처리를 통해 시각적으로 중요도가 높은 영역을 모델 입력에 반영하고, 그 결과를 바탕으로 “각막 표면 혼탁”, “결막 충혈”, “눈물 자국 및 텔 변색” 등 구체적인 시각적 단서를 서술한다. 또한 JSON, Markdown, 챗봇형 응답 등 다양한 포맷을 통해 진단명·증상·관리 방법을 구조적으로 제시함으로써, 보호자가 AI 판단 과정을 추적·이해할 수 있는 환경을 제공한다. 이는 의료 영역에서 AI 도입 시 흔히 제기되는 “블랙박스 문제”를 완화하고, 수의사–보호자–AI 간 협업 구조를 설계하는 데 기여할 수 있다.

셋째, 사용성 측면에서의 접근성 및 포용성 제고라는 기대 효과가 있다. 사용자는 모바일·웹 환경에서 “시작하기 → 이름 입력 → 사진 업로드 → 결과 확인 → 챗봇 상담”의 단순한 플로우만으로 서비스를 이용할 수 있으며, 별도의 회원가입이나 복잡한 설정 과정 없이 직관적인 UI를 통해 진단 정보를 획득할 수 있다. 이는 IT 활용 능력이 낮은 고령층이나 동물병원 접근성이 낮은 지역 보호자에게 특히 유리하며, 반려동물 헬스케어 영역에서 존재하는 정보·지역 격차를 줄이는 데 일정 부분 기여 할 수 있다.

넷째, 본 프로젝트는 VLM·LoRA·RAG/Web Search를 통합한 멀티모달 헬스케어 응용 사례로서 학술적·교육적 가치도 가진다. 제한된 GPU 자원과 데이터 환경에서 Instruction Dataset 설계, ROI 기반 이미지 전처리, Low-Rank Adaptation을 결합해 실험을 수행한 경험은, 향후 유사 도메인(피부 질환, 치과 질환, 고양이 질환 등)에 VLM을 적용할 때 재사용 가능한 실무적 지침을 제공할 수 있다. 또한 학생·연구자가 실제 서비스 수준의 파이프라인을 구축·운영하는 데 참고할 수 있는 구체적인 사례로 활용될 수 있다.

7.2 프로젝트 한계점

7.2.1 데이터 제약

가장 큰 한계는 이미지 품질과 도메인 특성을 둘러싼 데이터 제약이다. 본 연구에서는 공신력 있는 공개 데이터셋인 AI-Hub 반려동물 안구 질환 데이터를 기반으로 학습을 수행하였으나, 원천 이미지 중 상당수가 실제 보호자 촬영 환경과 유사한 저해상도·블러·노이즈를 포함하고 있다. 학습 전 단계에서 데이터 필터링 및 ROI 기반 전처리를 수행했음에도 불구하고, 일부 이미지는 초점이 안구가 아니라 털이나 주변 사물에 맞추어져 있거나, 조명이 과도하게 강하거나 어두워 병변이 충분히 드러나지 않았다.

또한 임상 라벨은 존재하지만, 세부 증상 수준의 정답 레이블(예: "각막 표면 함몰", "눈물점 폐쇄")은 별도로 제공되지 않아, 모델이 학습하는 텍스트 설명의 일부는 VLM 기반 자동 생성에 의존할 수밖에 없었다. 이는 미세 병변을 구분하는 데 필요한 고품질 "시각·텍스트 페어 데이터"가 부족하다는 의미이며, 특히 안검염·안검내반증·유루증과 같이 시각적으로 유사한 질환 간 혼동을 완전히 해소하지 못한 원인으로 작용하였다. 요약하면, 데이터의 절대적 양보다 질과 어노테이션의 정밀도 측면에서 제약이 있었고, 이는 모델 성능 상한선을 제한하는 요인으로 작용하였다.



	
그림 3. 이미지 흐림, 근접 촬영으로 인한 안구 이미지 품질 저하 사례	그림 4. 초점 불량·근접 촬영으로 발생한 저화질 안구 이미지 사례

7.2.2 비용 제약으로 인한 실험 제한

두 번째 한계는 연산 자원 및 비용 제약으로 인한 실험 설계의 제약이다. 본 연구는 주로 단일 A100 80GB GPU 환경에서 수행되었으며, 모델 학습 1회에 약 10–11시간이 소요되는 등 개별 실험의 비용이 상당히 높았다. 이로 인해

- LoRA 랭크, 학습률, 스케줄러, 배치 크기 등 하이퍼파라미터 조합을 광범위하게 탐색하지 못하였고,
- Vision Encoder 와 LM Decoder 각각에 대한 어댑터 주입 위치·범위에 대한 체계적인 어플레이션 스터디를 수행하는 데에도 한계가 있었다.

또한 Multi-Image 입력 전략(원본 + 크롭)이나, 다양한 ROI 전략(단순 크롭 vs 패딩 크롭 vs 멀티 스케일) 등은 일부 시도 후 성능 저하가 관찰되었음에도, 충분한 반복 실험과 통계적 검증을 수행하기 어려웠다. 결과적으로, 본 보고서에서 보고하는 실험 결과는 주어진 자원 내에서 가장 실용적인 선택을 도출한 것이며, 이론적으로 가능한 최적 조합을 찾았다고 보기에는 어렵다는 한계가 존재한다.

7.2.3 수의학 분야 전문성 결여

세 번째 한계는 수의학 분야의 전문 지식을 충분히 갖추지 못했다는 점이다. 본 프로젝트는 컴퓨터공학·인공지능 전공자를 중심으로 수행되었기 때문에

- 질환별 임상적 중요도
- 실제 진료 현장에서의 의사결정 흐름
- 특정 증상이 갖는 병리학적 의미 등을 정교하게 반영하는 데 한계가 있었다.

예를 들어, Instruction Dataset을 설계할 때 질환별 증상 설명과 관리 방법은 문헌·온라인 자료·LLM 보조를 통해 구성되었지만, 실제 임상에서 사용되는 표현·우선순위·주의사항이 완벽히 반영되었다고 보기는 어렵다. 또한 혼동이 잦은 클래스에 대해 수의사가 수작업으로 세부 라벨을 재정의하거나, 오분류 사례를 검토하여 “임상적으로 허용 가능한 오차 범위”를 설정하는 과정이 충분히 이루어지지 못했다. 이는 향후 실제 의료 현장 적용 시 전문의 검증 및 피드백 루프를 반드시 추가로 구축해야 한다는 점을 시사한다.

7.3 향후 연구 및 서비스 확장 계획

앞서 언급한 기대 효과와 한계를 바탕으로, 향후 연구 및 서비스 확장 방향은 크게 데이터·모델 측면의 고도화, 서비스 기능 확장, 임상 적용을 위한 검증·협업 체계 구축의 세 축으로 정리할 수 있다.

첫째, 데이터 및 모델 측면의 고도화가 필요하다. 우선, 수의사와의 협업을 통해 고해상도 임상 사진과 정밀 어노테이션(질환 단계, 위치, 중증도, 동반 질환 등)을 포함한 고품질 전용 데이터셋을 구축하는 것이 중요하다. 이 과정에서 액티브 러닝(active learning)이나 하드 예제 마이닝(hard example mining) 전략을 도입하면, 모델이 혼동하는 사례를 중심으로 효율적으로 데이터를 확장할 수 있다. 또한 견종, 연령, 기저 질환, 촬영 환경과 같은 메타데이터를 함께 수집하여 멀티모달 입력으로 활용한다면, 도메인 일반화 능력과 공정성(fairness)을 동시에 개선할 수 있을 것으로 기대된다.

모델 구조 측면에서는, 현재 사용 중인 단일 VLM 구조에 더해 전통적인 CNN/ViT 기반 특화 분류기와의 양상블, 혹은 질환 분류와 보고서 생성을 분리한 이중 헤드(dual-head)아키텍처를 실험함으로써, 진단 정확도와 설명 품질을 각각 최적화하는 전략을 고려할 수 있다. 또한 RAG 및 Web Search 모듈과의 통합을 고도화하여, 드문 질환이나 업데이트가 잦은 치료 가이드라인에 대해서도 최신 정보를 반영할 수 있는 근거 기반 설명 시스템으로 발전시키는 것이 장기적인 목표이다.

둘째, 서비스 기능 수준에서의 확장을 계획하고 있다. 현 단계에서는 안구 질환을 중심으로 시스템을 구축하였으나, 향후 반려견 피부 질환, 반려묘 안구·피부 질환, 치아·체형·관절 등 다른 신체 부위로 대상 질환군을 확대하는 것이 가능하다. 동일한 VLM 기반 아키텍처를 유지하되, 입력 모달과 Instruction 템플릿을 상황에 맞게 조정함으로써 재사용성을 높일 수 있다.

또한, 진단 결과를 단일 시점의 판단으로 제시하는 것을 넘어, 시간 축을 고려한 건강 리포트 기능(예: "지난 6개월간 촬영된 사진 기준 상태 변화 추세 시각화")을 추가하면, 보호자가 장기적인 관리 흐름을 이해하는 데 도움을 줄 수 있다. 위치 기반 서비스를 결합하여 주변 동물병원 추천, 진료과 특화 병원 안내, 진료 후 처방 내용 기록·요약을 지원하는 등, 실제 진료 과정 전후를 아우르는 통합 헬스케어 플랫폼으로 확장하는 것도 가능하다.

셋째, 임상 적용을 위한 검증 및 협업 체계 구축이 필요하다. 연구 단계의 모델이 실제 진료 현장에서 사용되기 위해서는,

- 수의사 대상 파일럿 테스트
- 실제 병원 데이터에 대한 외부 검증(validation)
- 임상 윤리 및 법적 규제 준수

등을 포함한 다각도의 검증이 필수적이다. 향후에는 수의사가 직접 모델 출력을 검토·수정하고 그 결과가 다시 학습 데이터로 환류되는 휴먼 인 더 루프(human-in-the-loop)구조를 도입하여, 시간이 지날수록 모델 성능이 점진적으로 개선되는 자가 개선(self-improving) 시스템을 구축할 계획이다.

마지막으로, 개인정보 보호와 데이터 보안 측면에서 온디바이스 추론 또는 연합학습(Federated Learning)과 같은 프라이버시 보존형 학습 기법을 도입하는 것도 장기적인 연구 과제로 제시할 수 있다. 이를 통해 사용자의 민감한 이미지 데이터를 중앙 서버로 수집하지 않으면서도, 다양한 사용 환경에서 모델을 지속적으로 개선하는 방향을 모색할 수 있을 것이다.

종합하면, 본 프로젝트는 제한된 환경에서 VLM 기반 반려동물 질환 진단 보조 시스템의 가능성을 실증한 초기 연구로서 의미를 가지며, 향후 데이터·모델·서비스·임상 협업 측면의 후속 연구를 통해 실제 현장에서 활용 가능한 수준의 설명 가능한 반려동물 헬스케어 플랫폼으로 발전할 잠재력을 지니고 있다.

8. 역할 분담 및 참고 문헌

8.1 역할 분담

1) 나영석

[시스템 및 인프라 개발]

- Web Client 개발 및 Runpod Serverless 백엔드(Backend) 환경 구축

[데이터셋 구축 (API 활용)]

- VLM API를 활용한 증상 추출 데이터셋 구축
- LLM API를 활용한 챗봇 학습 데이터 생성

2) 문보라

[이미지 분류 모델링]

- Resnet50 및 CLIP 모델을 활용한 Classification 구현모델 성능 평가 및 분석

[프로젝트 문서화]

- 프로젝트 최종 보고서 정리

3) 민하은

[챗봇 서비스 및 파이프라인 통합]

- **Qwen3-VL 기반 챗봇 모듈 구현** : 텍스트 및 이미지 처리가 가능한 챗봇 코어 개발
- **검색 및 지식 연동** : 챗봇 모듈에 RAG 및 Web Search 기능 연동 (Module Integration)
- **시스템 통합** : 전체 End-to-End 서비스 파이프라인 구축 및 최적화

4) 이호철

[데이터셋 및 파이프라인 구축]

- **데이터셋 구축 및 가공** : LLM API를 활용한 설명형 데이터 생성 및 전체 데이터 전처리
- **검색 시스템 개발** : RAG(Retrieval-Augmented Generation) 및 Web Search 파이프라인 구현

[AI 모델 학습 및 최적화]

- **VLM 파인튜닝** : Qwen3-VL-8B 모델 LoRA 파인튜닝 수행 및 성능 검증 (Validation)
- **객체 탐지 및 전처리** : Yolo 모델 파인튜닝 및 이미지 크롭(Crop) 모듈 구현

8.2 참고 문헌

- [1] P. Lewis *et al.*, “Retrieval–Augmented Generation for Knowledge–Intensive NLP Tasks,” *arXiv preprint arXiv:2005.11401*, version 4, 2021. 2005.11401v4
- [2] E. Hu *et al.*, “LoRA: Low–Rank Adaptation of Large Language Models,” *arXiv preprint arXiv:2106.09685*, version 2, 2021. 2106.09685v2
- [3] Qwen Team, “Qwen3–VL Technical Report,” *arXiv preprint arXiv:2511.21631*, version 2, 2025. 2511.21631v2
- [4] Z. Gan, L. Li, C. Li, L. Wang, Z. Liu, and J. Gao, “Vision–Language Pre–training: Basics, Recent Advances, and Future Trends,” *arXiv preprint arXiv:2210.09263*, version 1, 2022. 2210.09263v1