|  |
| --- |
| **1. 주제**  실시간 화면 번역 프로그램 개발  **분반, 팀, 학번, 이름**  나분반, 9팀, 20251772, 유의선 |

|  |  |
| --- | --- |
| **2. 요약**  목표: 아이패드 환경에서 화면을 실시간으로 번역하는 프로그램 개발  핵심 기능: 화면에 일정 영역을 지정하여 지정된 시간마다 영역 내의 텍스트를 인식, 텍스트가 변화함에 따라 새로운 번역을 진행해 번역본을 출력한다.  기대 효과  정보 습득 속도를 일일이 복사/캡처 후 번역기로 옮기는 비효율적인 과정에서 벗어나면서 향상시킬 수 있다.  텍스트 변화가 있을 때만 번역 엔진을 호출하기 때문에 API 호출 비용을 최소화할 수 있다.  채팅 뿐만 아니라 스크롤 형식으로 보는 웹툰 등의 만화나, 영상의 자막 등을 번역하는 데에도 적합하다. | **3. 대표 그림**  **프로그램 실사용 예시 – chess.com에서 외국인과의 대화에서 이용할 수 있다.** |

|  |
| --- |
| **4. 서론**  현대의 디지털 환경에서는 다양한 언어로 된 콘텐츠를 접할 기회가 늘어나게 되었다. 예를 들면 온라인으로 다른 국적의 사람과 대화하거나, 다른 나라의 음악, 또는 사이트를 이용하는 경우 등이 있다. 그러나 일반적으로 이를 번역하려고 하면 번역하고자 하는 내용을 복사 또는 캡처해서 번역기에 붙여넣는 과정을 거치거나, 웹 브라우저의 전체 페이지 번역 기능을 이용해야 한다. 이러한 기존의 방식은 간단한 번역에는 적합하나, 번역할 페이지가 많은 경우 절차가 복잡하고 시간이 많이 소요되는 한계가 발생한다. 필자는 온라인 체스를 하면서 다양한 국적의 상대방을 만나고, 대화하면서 이 비효율적인 과정에 대해 불편함을 느껴 아이디어를 가지게 되었다.  기존 번역 방식의 가장 큰 문제점은 번역 과정의 비효율성이다.  1. 번역하고자 하는 내용이 많을 경우, 사용자는 매번 캡처 또는 복사 후 번역기 앱을 실행해 붙여넣는 과정을 반복해야한다.  2. 전체 페이지 번역 기능은 불필요한 부분까지 번역하여 번역속도를 늦추고, 어색한 번역이 많아지면 오히려 정보의 가독성을 떨어뜨린다.  이는 생산성과 사용자 경험 측면에서 모두 비효율적이다. |

|  |
| --- |
| **5. 본론**  **시스템 구성도**  본 프로젝트는 사용자가 지정한 화면 영역의 텍스트를 실시간으로 감지하고 번역하여 원본 텍스트 위에 오버레이 형태로 즉시 보여주는 iPadOS 애플리케이션이다. 특히 온라인 게임 채팅, 실시간 강의 자료, 동영상 자막, 또는 스크롤하여 감상하는 만화 등 콘텐츠의 흐름이 중요한 상황에서 사용자가 복사 또는 캡처와 앱 전환 없이 즉각적으로 정보를 습득할 수 있게 한다. 시스템의 핵심 동작 과정은 다음의 루프와 같다.  영역 지정 -> 주기적 캡처 및 텍스트 인식 -> 번역 -> 텍스트 출력  사용자가 번역을 원하는 영역을 지정하면, 해당 영역을 설정된 시간 간격으로 지속적으로 캡처하고, 텍스트를 추출하여, 텍스트의 변화가 있다면 번역하여 결과를 출력한다. 변화 감지 과정을 통해 불필요한 번역 및 텍스트 출력을 최소화하여 배터리 소모를 최소화한다.  개발을 위해선 Apple의 공식 언어인 Swift를 사용해야 iPadOS 환경에서의 안정성과 성능을 확보할 수 있을 것이다. 또한, 광학 문자 인식으론 Apple에서 제공하는 Vision Framework를 활용하여 빠르게 이미지 속 텍스트를 추출할 수 있다. 기계 번역은 Apple의 Translation Framework를 주 번역 엔진으로 사용한다. 이것은 Vision Framework와 함께 기기 내에서 수행되기 때문에 외부 API 호출 없이 즉각적인 번역이 가능하다.  화면 캡처는 UIGraphicsImageRenderer를 활용해 수행한다. 사용자가 지정한 특정 영역의 화면을 캡처해 시스템 전체 화면이 아닌 필요한 부분만 효율적으로 가져와 처리 부하를 줄일 수 있어야한다.  핵심 로직 구현  실시간 처리 루프: Swift의 Timer 객체를 사용하여 주기적으로 화면 캡처 및 OCR을 수행하는 실시간 루프를 구현한다. 사용자가 번역 모드를 활성화하면 타이머가 시작되고, 비활성화하면 종료되어 시스템 부하를 관리한다.  변화 감지 최적화: OCR로 추출된 텍스트를 변수에 저장하고 다음 루프에서 추출된 텍스트와 비교(String diffing)한다. 두 텍스트가 다를 경우에만 Translation Framework를 호출하여 번역을 진행하게 하여 정적인 화면에서의 불필요한 연산을 방지한다.  사용자 인터페이스(UI) 구현  영역 선택 인터페이스: 사용자가 화면 위에서 드래그하여 영역을 지정하고, Trans ON/OFF 버튼과 Tick Rate(캡처 시간 간격)을 표시한다.  동적 오버레이 뷰: SwiftUI의 .overlay() 수정자를 활용하여 원본 콘텐츠 위에 번역 결과를 겹쳐 표시한다. 번역된 텍스트의 가독성을 위해 반투명 배경과 폰트의 테두리색을 지정한다.  향후 개발 방향  언어 설정: 번역 가능한 언어 확장(일본어, 중국어 등) / 한국어 번역 이외에도 다른 언어 지원  번역 결과 활용: 번역된 텍스트를 클립보드에 복사하거나 다른 앱으로 공유하는 등 부가적인 편의 기능을 추가 |

|  |
| --- |
| **6. 결론**  본 프로젝트는 iPad 사용자들이 실시간으로 변화하는 화면 콘텐츠를 번역하는 과정에 있어서 기존의 비효율적인 과정을 개선하고자 제안되었다. 본 시스템은 iPadOS 환경의 Vision Framework(OCR)와 Translation Framework(번역)를 활용하여 사용자가 지정한 영역의 텍스트 변화를 감지하고 초 단위로 실시간 번역 결과를 원본 화면 위에 오버레이하여 표시할 것이다. 변화 감지를 통해 최적화하여 배터리 효율성을 확보하며, 궁극적으로 정보 습득의 속도와 집중도를 획기적으로 향상시키는 것을 목표로 한다.  향후 계획  1. 핵심 기능 프로토 타입 개발: SwiftUI 기반 영역 선택 및 UI 오버레이 기능 구현  Vision/Translation Framework 연동 및 텍스트 변화 감지 로직 구현  2. 실시간 루프 성능 최적화: Timer 간격 테스트를 통한 안정적인 Tick Rate 확보  3. 사용자 편의 기능 구현: 번역 결과 텍스트 복사/공유 기능 구현  설정화면 (언어 선택, 오버레이 스타일 조정) 구현  4. 최종 안정성 테스트: iPad 모델 및 iOS 버전에서의 호환성 및 부하 테스트  5. 오픈소스 커뮤니티 문서화: 상세 기술 문서 및 설치/사용 가이드(README) 작성 및 공개 |

.