# Kexxu Eye

여기에서 Kexxu 안구 추적 안경을 구매하세요. 배터리 수명이 6 시간인 완전 휴대형 안구 추적 안경입니다.
시선 추적 안경은 누군가가 보고 있는 것을 기록합니다. 마케팅, 제품 연구, 사용자 경험 디자인, 스포츠 분석 등에 Kexxu 시선 추적 안경을 사용하세요.
당사의 웨어러블 시선 추적기는 하나의 시선 추적 카메라로 눈을 기록하고 다른 카메라로 장면을 기록하여 시선을 기록할 수 있습니다.
카메라에 인공지능이 달려 눈을 기록해 동공의 중심을 실시간으로 파악한다.
동공 위치를 사용하여 장면을 기록하는 카메라에 십자선이 그려집니다. 이렇게 하면 장면에서 사람이 어디를 보고 있는지 실시간으로 확인할 수 있습니다.
시선 추적 기록 후 추가 분석을 위해 기록을 Kexxu Cloud 로 보낼 수 있습니다. 여기서 히트 맵, 시선 플롯, 관심 영역 생성 등을 생성할 수 있습니다!

# Description
# Kexxu Eye

휴대성이 뛰어나고 저렴한 시선 추적 안경입니다.

고객이 무엇을 보고 있나요? 최근 광고에서 가격표를 발견했나요? 기존 패키지 디자인보다 새로운 패키지 디자인을 더 많이 보셨나요? 그리고 귀하의 매장은 어떻습니까? 고객이 들어왔을 때 가장 먼저 눈에 띄는 것은 무엇입니까? Kexxu 아이 아이트래킹 안경을 통해 사람들이 정말로 관심을 갖는 것이 무엇인지 알아보세요.

Kexxu 아이트래커 히트맵

## Specification

**완벽한 휴대성**
**배터리 수명 6 시간**
**2 도 정확도**
**초당 20 프레임**
**모든 처리는 아이트래커에서 실시간으로 이루어짐**
**WiFi 를 통한 실시간 미리보기로 손쉬운 보정**
**16 시간의 로컬 저장 공간**
**2GB 의 무료 클라우드 저장 공간**

## 클라우드 도구

Kexxu Eye 에는 편리한 클라우드 도구가 포함되어 있습니다. 컴패니언 앱에서는 아이트래커의 녹음 내용을 kexxu 편집 스튜디오로 보낼 수 있습니다. 고급 최신 AI 도구를 사용하면 다음을 만들 수 있습니다.

**AI** 기반 관심 영역 추적 및 분석
집중 시간 측정
동적 히트 맵은 머리를 움직일 때도 작동합니다!
동적 시선 플롯은 이미지를 안정화하여 단속적 움직임의 시선 플롯을 생성합니다.

## 앱

실시간 미리보기 및 보정을 위한 앱은 다음과 같습니다.

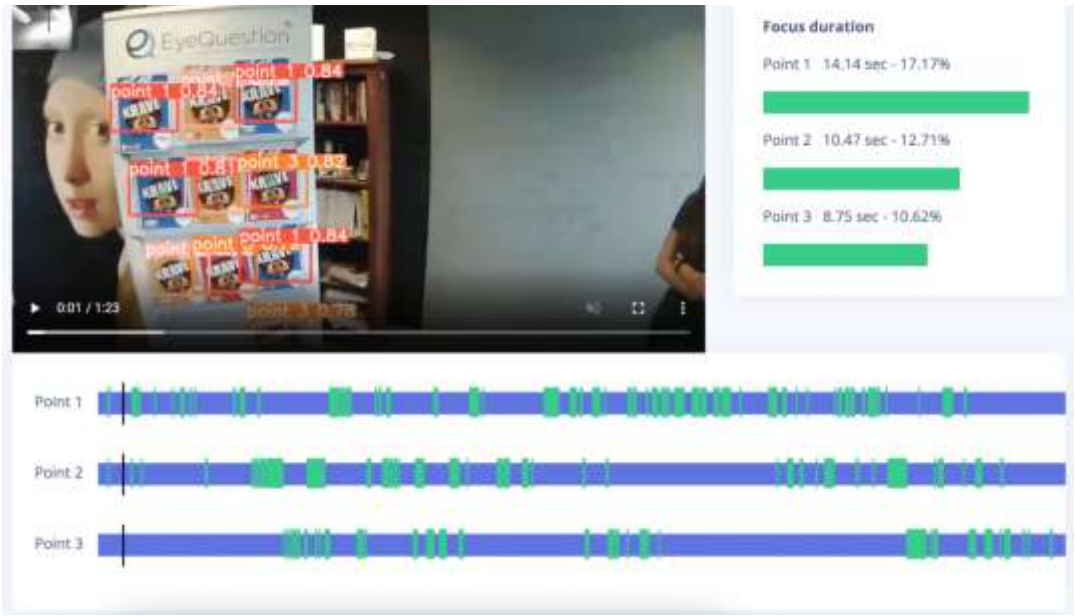
안드로이드
**iOS**

**시선 추적 연구의 주력 제품입니다.**

**온라인 시선 추적 소프트웨어 사용.**

### AI 학생 추적 소프트웨어

당사의 시선 추적 하드웨어와 클라우드에서 실행되는 시선 추적 소프트웨어를 결합하면 최신 AI 동공 추적 소프트웨어를 활용하여 중요한 시선 지속 시간 통계를 측정할 수 있습니다. 시선 지속 시간 또는 체류 시간은 참가자가 무언가를 바라 본 시간을 나타냅니다.
일반적으로 상대적인 체류 시간에 관심이 있을 것입니다. 파란색 또는 빨간색 포장의 체류 시간이 더 큽니까?
editor.kexxu.com을 사용하면 다양한 관심 영역을 비교하는 체류 시간 비교 다이어그램을 만드는 것이 매우 쉽습니다.
계정을 생성하면 세 가지 관심 영역에 대해 체류 시간이 계산되는 예시 영상이 제공됩니다.



## 완벽한 휴대성

아이트래커는 안경과 안경에 부착된 처리 장치로 구성됩니다. 모든 처리는 이 휴대용 장치에서 실시간으로 이루어집니다. Kexxu 클라우드 편집기에 데이터를 업로드하기 전에 최대 16시간의 녹음 시간을 저장할 수 있습니다.
6시간의 배터리 수명으로 단 한 번의 충전으로 하루 종일 녹음이 가능합니다. 배터리를 쉽게 꺼내서 교체하거나 충전할 수 있습니다. 처리 장치 뒷면에는 현재 배터리 비율이 표시됩니다.


## 관심 영역 시선 추적

시선 추적 무료 클라우드 분석 도구를 사용하면 시선 추적 기록에 관심 영역을 표시할 수 있습니다. 당사의 클라우드 도구는 관심 영역에 대해 알고 싶은 모든 것에 대한 통찰력을 제공합니다. 예를 들어 관심 영역당 응시 시간이나 집중하는 시간 등이 있습니다.
집중 시간은 누군가가 환경에서 무언가를 알아차리는 데 걸리는 시간을 측정합니다. 타이머는 관심 영역이 시야에 들어올 때 시작되고, 사람이 관심 영역을 직접 바라볼 때 멈춥니다. 예를 들어 대시보드에 경고등이 켜졌다고 가정해 보겠습니다.
editor.kexxu.com을 사용하면 AI 지원을 통해 집중 비교 다이어그램을 만드는 것이 매우 쉽습니다. 여기에서 관심 영역을 사용하여 집중하는 시간을 측정하는 방법에 대해 자세히 알아 보세요 .


## 히트맵

온라인 시선 추적 인터페이스에서는 시선 추적 기록의 열 지도를 만드는 것이 매우 쉽습니다. 히트 맵은 시선 추적 데이터에 대한 빠른 통찰력을 얻는 가장 쉽고 빠른 방법입니다.
인공 지능을 사용하여 머리 움직임을 보상합니다. 이러한 방식으로 우리는 착용 가능한 시선 추적 안경을 사용하여 환경을 걸을 때에도 초점 포인트의 안정적인 열 지도를 생성할 수 있습니다.
editor.kexxu.com 에서 버튼 하나만 누르면 시선 추적 실험의 히트맵을 만들 수 있습니다 .

## 시선 플롯

게이즈 플롯은 단속운동과 단속운동의 순서를 시각화할 수 있습니다.
모든 사람은 추적과 단속운동이라는 동일한 두 가지 방식으로 눈을 움직입니다. 추적이란 물체를 따라가거나 자신을 움직이면서 물체에 시선을 고정하는 것을 말합니다. 단속운동은 눈을 하나의 추적 위치에서 다른 추적 위치로 이동합니다. 단속운동은 초당 3~7회 발생하며 시선 플롯으로 시각화됩니다.
editor.kexxu.com 에서 버튼 하나만 누르면 시선 추적 실험의 시선 플롯을 만들 수 있습니다 .

## 기본 시선추적 실험

테스트 대상에게 Kexxu 안구 추적 안경을 착용하도록 설득하여 실험을 시작합니다. 안경은 Kexxu Devices 앱에 연결되어 녹화된 비디오와 피사체가 보고 있는 위치를 미리 볼 수 있습니다. 먼저 아이트래커를 보정하는 것부터 시작합니다.
시선 추적기를 보정하려면 시선 추적 AI가 작업을 수행할 수 있도록 AI 카메라 창을 눈 위에 완벽하게 맞도록 조정하세요. 그런 다음 피험자가 시야 중앙에 있는 지점을 보도록 합니다. 시선 교차는 처음에는 해당 위치를 가리키지 않으므로 앱의 버튼으로 시선 교차를 조정하여 십자가가 테스트 대상이 보고 있는 지점에 정확하게 위치하도록 합니다.
일부 눈의 경우 감도가 다릅니다. 이것이 무엇을 의미합니까? 피사체가 왼쪽과 오른쪽의 한 지점을 바라보게 하면 시선 크로스가 대상을 초과하거나 미달할 수 있습니다. 가자십자가가 중앙 뿐만 아니라 측면에서도 정확하게 표적에 맞도록 앱에서 감도를 조정하세요. 위아래로 동일하게 수행하십시오. 감도는 일반적으로 약간만 조정하면 됩니다.
그런 다음 녹음을 누르면 눈 움직임 기록이 시작됩니다. 이제 중요한 것은 피험자에게 지시를 내

리는 것입니다. 눈의 움직임과 시선의 위치는 지시에 따라 많이 다릅니다. 예를 들어, 가장 저렴한 제품을 찾으라고 요청하는 것은 최고 품질의 제품을 찾으라고 요청하는 것과는 다른 시선 플롯을 제공합니다.
실험이 끝나면 녹음 중지를 누를 수 있습니다. 녹음 내용은 시선 추적 안경에 저장되지만 "클라우드에 업로드"를 누를 수 있는 앱의 목록에 표시됩니다. 비디오가 클라우드에 업로드되면 클라우드 편집기로 이동하여 시선 플롯, 히트 맵을 생성하고 관심 영역을 표시하여 통계적으로 비교할 수 있습니다.

## Kexxu Eye

완전히 휴대 가능하고 독립형
2 도 정확도
6 시간의 배터리 수명
기기 보관 시 16 시간
무료 2GB 의 클라우드 스토리지

## 방법: Python 을 사용하여 Kexxu Eye 출력 처리

우리는 Kexxu 눈 데이터를 처리하기 위한 다양한 고급 클라우드 도구를 제공하지만 간단한 Python 스크립트를 사용하여 시선 추적 출력 데이터를 직접 쉽게 구문 분석할 수도 있습니다. 방법을 알아보려면 이 튜토리얼을 확인하세요!
여기서 더 읽어보세요

```
loss = []
with open(path, "r") as f:
  lines = f.readlines()
  for line in lines:
    parts = line.split(" ")
    if parts[0].endswith("/eyetracking"):
      js = json.loads(" ".join(parts[2:]))
      xs = js["pupil_rel_pos_x"]
      ys = js["pupil_rel_pos_y"]
      x = 640 - int((xs) * 640.0)
      y = 360 + int((ys) * 360.0)
      loss.append((x, y))
return loss
```