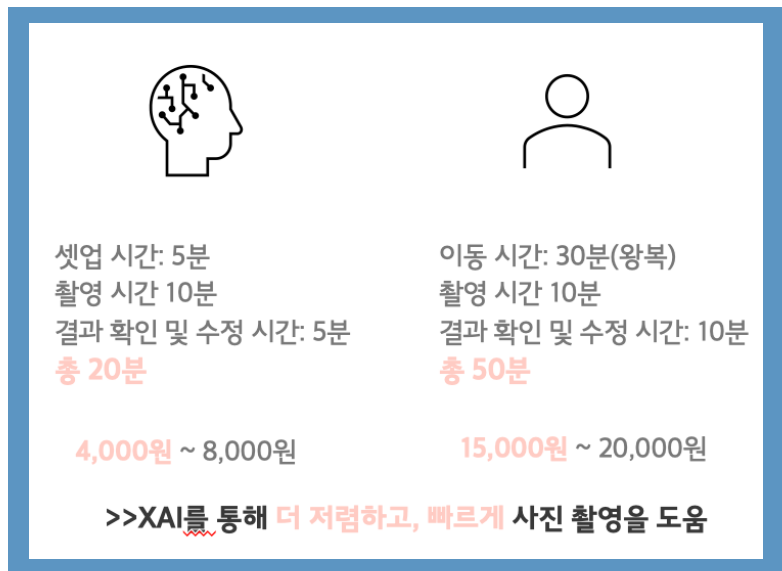


XAI 사진사와 함께하는 인생샷

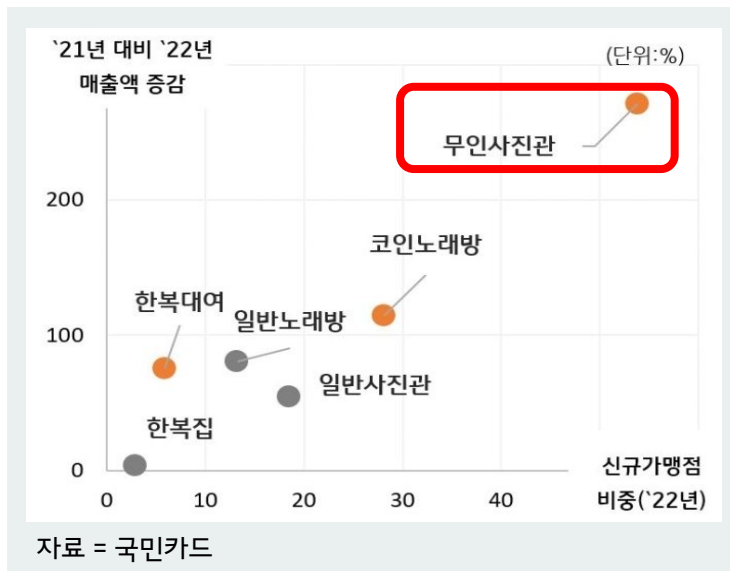


소프트웨어학과
20201659 박은지

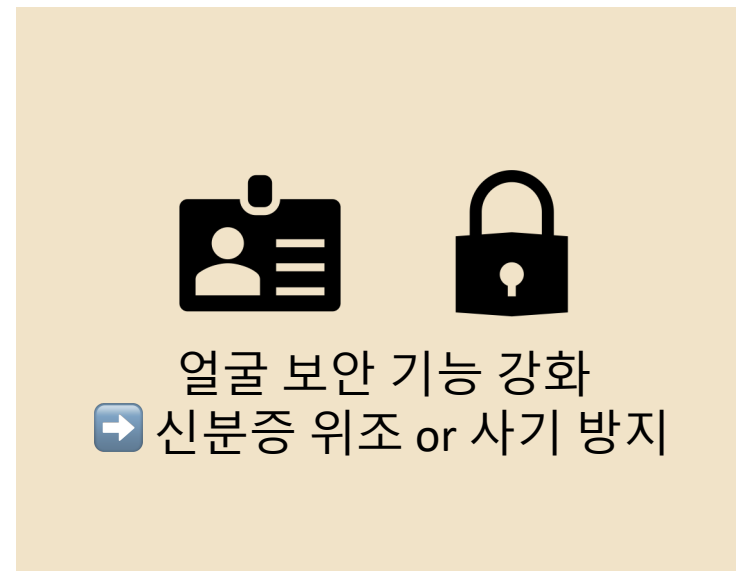
Part 1 배경 및 필요성



1. 시간과 비용 최소화

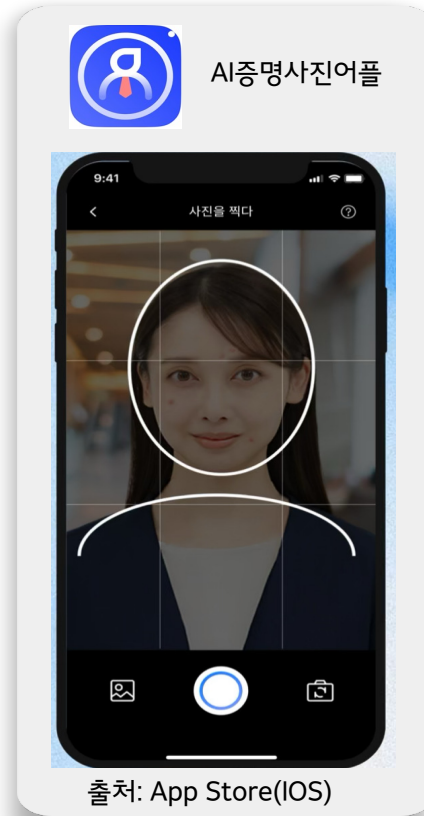
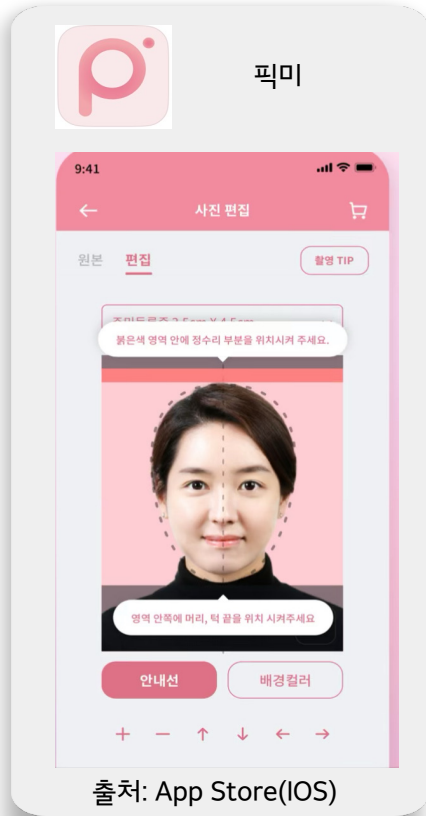
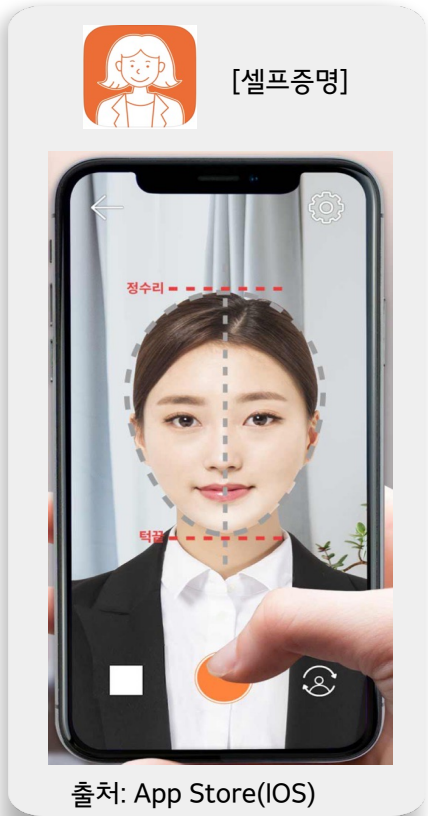


2. 현재 시장 가치



3. 사회적 파급효과

Part 2 관련 기술 조사(기존 서비스)



[기존 서비스의 공통점]

1. 규격 선택
2. 셀프 촬영(머리끝과 턱 끝 맞춰 촬영)
3. 셀프 보정 및 배경 편집

[기존 서비스의 단점]

1. 제한적인 가이드
2. 안내선에 따라 수동적인 자세 조정

[차별 서비스]

머리끝과 턱 끝 가이드 +
고개의 각도 - 좌우

인식 후 설명가능한 AI의 실시간 피드백

(Driving Monitoring System)

OpenCv 및 Dlib를 활용한 DMS 개발

Moon, S.-H., Lee, J.-H., Yun, S.-J., So, S.-M., & Baek, S.-W. (2022).

Development of DMS(Driving Monitoring System) using OpenCv and Dlib.

Proceedings of the Korea Information Processing Society Conference, 919-921.

OpenCv, Dlib를 활용하여 실시간으로

운전자의 상태를 촬영하여

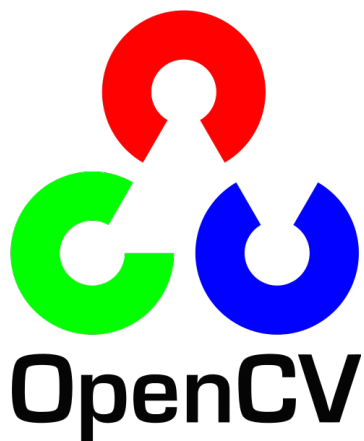
눈의 크기, 고개 숙임 등을 감지하여 졸림, 시선 부주의 판단 시

알람을 통하여 운전자에게 경고를 주는 시스템을 개발



<그림 1> FaceLandMark 기술의 특징점 추출.

1. OpenCV

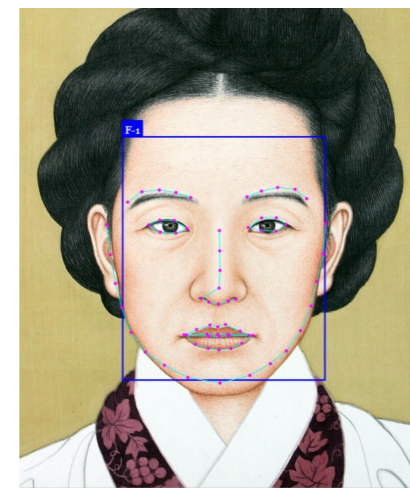


컴퓨터 비전 및 이미지 처리에
사용되는 오픈 소스 라이브러리

- 이미지 및 비디오 캡처 처리
- 객체 인식 및 추적
- 머신 러닝 및 딥 러닝 지원

- 얼굴 감지
- 얼굴 랜드마크 검출
- 얼굴 인식
- 얼굴 표정 인식

2. Face-api.js



브라우저 환경에서 얼굴 인식
및 랜드마크 검출 수행하는
JavaScript 라이브러리

Step 1 이미지 분석 OpenCV, FaceAPI

- 얼굴 검출
- 얼굴 특징 점(68개) 검출
- 고개의 기울임 정도 파악



Step 2 XAI 설계

- 모델의 촬영 위치 결정 기준 이해
- 모델이 사용한 입력데이터에 대한 설명 제공
- XAI가 사용자에게 시각적, 청각적 가이드 제공.

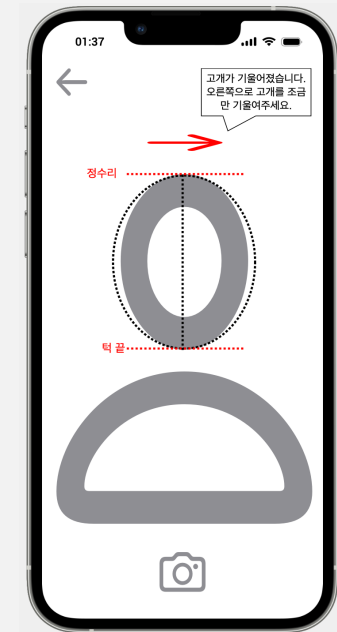
고개가 기울어졌습니다.
오른쪽으로 고개를 조금
만 기울여주세요.

고개가 기울어졌습니다.
앞쪽으로 고개를 조금만
숙여주세요.

안경에 빛이 비칩니다.
안경을 똑바로 써주십시
오.

Step 3 인터페이스 설계

- 사용자 인터페이스 설계
- 앱의 로직 및 코드 연결
- 디버깅 및 테스트





<그림 1> FaceLandMark 기술의 특징점 추출.

- 얼굴의 특정 랜드마크를 탐지하여 얼굴의 방향과 기울기 판단
- face-api.js 라이브러리를 사용하여 얼굴을 감지하고, 랜드마크 추출

1. 얼굴 랜드마크 감지

```

async function detectFace() {
  const detections = await faceapi.detectAllFaces(video, new faceapi.TinyFaceDetectorOptions()).withFaceLandmarks();
  context.clearRect(0, 0, overlay.width, overlay.height);
  context.save();
  context.scale(-1, 1);
  context.translate(-overlay.width, 0);

  faceapi.draw.drawDetections(overlay, detections);
  faceapi.draw.drawFaceLandmarks(overlay, detections);

  context.restore();

  detections.forEach(detection => {
    const landmarks = detection.landmarks.positions;
    const explanation = isFrontalFace(landmarks);
    //const label = explanation.isFrontal ? "Frontal" : "Not Frontal";

    landmarks.forEach(point => {
      context.beginPath();
      context.arc(overlay.width - point.x, point.y, 2, 0, 2 * Math.PI);
      context.fillStyle = "green";
      context.fill();
    });

    provideFeedback(explanation.tilt);

    if (explanation.isHorizontal) {
      if (!isFrontal) {
        isFrontal = true;
        startTimer();
      }
    } else {
      if (isFrontal) {
        isFrontal = false;
        clearInterval(countdownIntervalId); // 타이머 중지
        timerElement.textContent = ''; // 타이머 초기화
      }
    }
  });
  context.restore();
}

```

2. 얼굴 방향 판단

```
function isFrontalFace(landmarks) {
  const leftEyeCenter = getMidpoint(landmarks[36], landmarks[39]);
  const rightEyeCenter = getMidpoint(landmarks[42], landmarks[45]);

  const slopeHorizontal = calculateSlope(leftEyeCenter, rightEyeCenter);
  const isHorizontal = Math.abs(slopeHorizontal) < 0.05;

  const tilt = slopeHorizontal > 0.05 ? "Left" : (slopeHorizontal < -0.05 ? "Right" : "None");

  console.log(`Horizontal: ${isHorizontal ? "Yes" : "No"}, Tilt: ${tilt}`);

  return {
    isHorizontal,
    tilt,
    //slopeHorizontal,
    explanation: `Horizontal: ${isHorizontal ? "Yes" : "No"}, Tilt: ${tilt}`
  };
};
```

- 얼굴의 랜드마크 기반으로 얼굴의 기울기 계산
- 왼쪽, 오른쪽 중 어느 방향인지 판단

3. 피드백 제공

```
function provideFeedback(tilt) {
  console.log(`Providing feedback for tilt: ${tilt}`);
  arrowElement.innerHTML = ''; // 화살표 초기화
  //guideVideoContainer.style.display = 'none'; // 가이드 영상 숨김
  switch (tilt) {
    case "Left":
      feedbackElement.textContent = "고개가 삐뚤어졌으니, 고개를 오른쪽으로 조금만 기울여주세요!";
      arrowElement.innerHTML = '-----'; // 왼쪽 화살표 표시
      //guideVideo.src = 'leftmoving.mp4'; // 왼쪽 가이드 영상 경로 설정
      //guideVideoContainer.style.display = 'block'; // 가이드 영상 표시
      break;
    case "Right":
      feedbackElement.textContent = "고개가 삐뚤어졌으니, 고개를 왼쪽으로 조금만 기울여주세요!";
      arrowElement.innerHTML = '←←←←←'; // 오른쪽 화살표 표시
      //guideVideo.src = 'rightmoving.mp4'; // 오른쪽 가이드 영상 경로 설정
      //guideVideoContainer.style.display = 'block'; // 가이드 영상 표시
      break;
    case "None":
      feedbackElement.textContent = "3초 동안만 정면을 유지해주세요!";
      break;
    default:
      feedbackElement.textContent = "3초 동안만 정면을 유지해주세요!";
  }
}
```

- 얼굴의 방향에 따라 사용자가 얼굴을 올바르게 조정할 수 있도록 피드백을 제공

향후과제(소프트웨어 관점)

사진 촬영 목적 추가

사진 촬영의 목적을 증명사진(정면)뿐만 아니라 사용자가 원하는 사진 목적에 따른 조건 추가

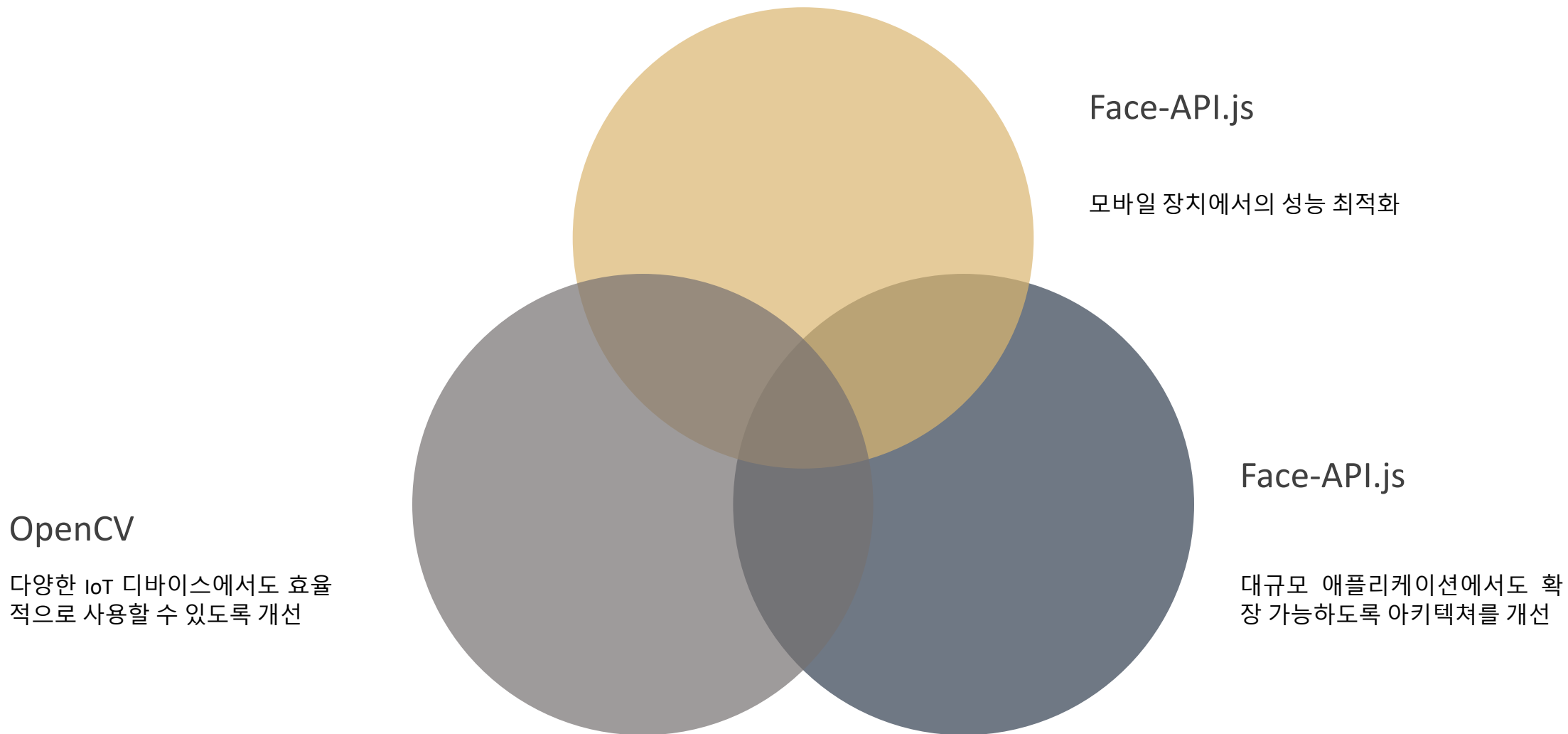
어깨, 안경 비침, 귀 인식 기능 추가

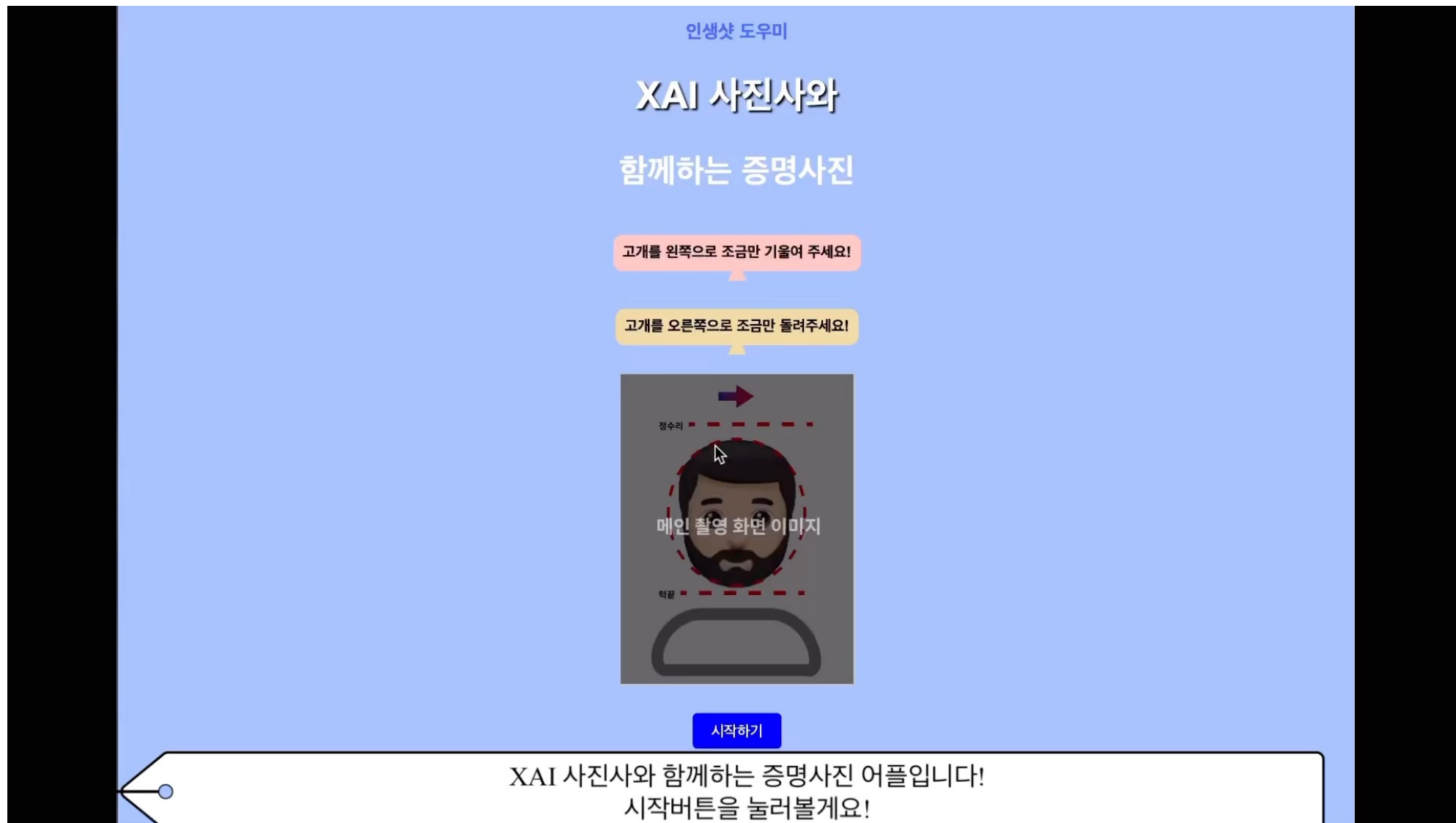
얼굴 뿐만 아니라 어깨, 안경 비침, 귀 인식도 인식하여 사용자가 원하는 사진 조건에 맞게 기능을 추가

용도에 따른 안내 창 추가

촬영 전 사진의 용도에 따른 규격 안내 창 추가

향후과제(연구 관점)







감사합니다.