

# 삼성화재 POC 스마트폰 보험 솔루션 최종 보고서

주식회사 맛앤멋

작성일 : 2021 . 11

# 목차

1

## POC를 통한 기술개발 개선 현황

- 1) Front-End
- 2) Back-End(AI학습)

2

## POC결과 보고

- 1) UI.UX기획 및 디자인(5월4주차~6월3주차)
- 2) 서버 및 Front 개발(6월1주차~9월2주차)
- 3) 학습 데이터 셋 확보(6월1주차~9월3주차)
- 4) 액정 인식률 및 판독률 고도화 개발 (6월3주차~9월3주차)

3

## TEST결과 보고

- 1) TEST 결과 및 추이 분석

4

## POC이후 추진 일정

- 1) API구축 및 기술상용화
- 2)사업화 진행 후의 보완 과제
- 3) 폴더블폰 DATA확보 및 연구 개발

# 1) Front-End

# 1. POC를 통한 기술개발 개선 현황



기존(ai.jamfon.co.kr)	
UI, UX	없음
촬영	Mirror 직접촬영 (갤럭시 모델 Focusing에 어려움 존재)
해상도	보통 수준
Noise	네비게이션바, 하단바 제거기능 없음
개발 내용	미러링 촬영
폴더블	<ul style="list-style-type: none"> <li>• 전면 촬영 가능</li> <li>• 내부 화면 촬영 불가능</li> <li>• 액정 이상 감지 불가능</li> </ul>

POC(coinchartapic.cafe24.com/jam/)
보험 가입 관련 UI,UX 제공
QR Code 촬영기법을 적용, 모든 기종의 Focusing이 용이하게 이루어져 쉬운 사진 촬영이 가능하도록 개선
해상도 개선
네비게이션바, 하단바 제거기능 개발(갤럭시폰)
<ul style="list-style-type: none"> <li>• 이미지 프로세싱을 통해 영상에서 휴대폰의 흰색면과, 휴대폰 윤곽선을 추출하여 액정 영역을 추출(10월중 개발완료 예정)</li> <li>• 이미지 프로세싱을 통한 과정의 추가로 화면에 색상으로 체크 할 수 있는 문제점들을 처리</li> </ul>
<ul style="list-style-type: none"> <li>• 전면 촬영 가능</li> <li>• 내부 화면 촬영 가능</li> <li>• Front-End에서 액정 이상 감지 가능</li> </ul>

### 기존(ai.jamfon.co.kr)

#### DATA 수집

- 파손된 스마트폰 직접 촬영
- 촬영자마다 사진의 Quality가 제각각
- 촬영기법의 일정한 기준이 부재
- 사용가능한 Data의 수량 부족

사용 가능 Data : 2,901장 반영  
(약 300여장을 Augemntation) 진행

Database가 없어 외장하드에 저장

#### Labelling

- 라벨링 Data : 객체 약 1만 개
- 깨짐, 찍힘, 크롬 벗겨짐, 기스(후면, 앞면), 기스(옆면)객체로 구분하여 라벨링

#### 학습

- Epoch : 150회
- 활용 모델 : Faster R-CNN
- 학습, 검증 셋 비율 : 8(2,320장):2 (581장)
- 모델 검증 Accuracy : 70~80%

### POC(coinchartapic.cafe24.com/jam/)

- Mirroring 촬영 Quality를 유지할 수 있는 기존 Solution이 존재 (ai.jamfon.co.kr)  
→ 높은 Quality의 Data 확보 가능

사용 가능 Data : 7,600여 장 중 노이즈를 제거된 4,664장 활용

(쥬맛앤멋 자체 Database 구축(아마존))

- 라벨링 DATA : 약 100,000개 (깨짐 객체 : 65,161개, 실금 객체 : 38,161개)
- 이 중 깨짐 객체만을 학습하여 판단 정확성을 높임

- Epoch : 1,000회
- 활용 모델 : Faster R-CNN
- 학습, 검증 셋 비율 : 8(3,731장):2(933장)
- 모델 검증 Accuracy : 98.42%



1. 초기화면



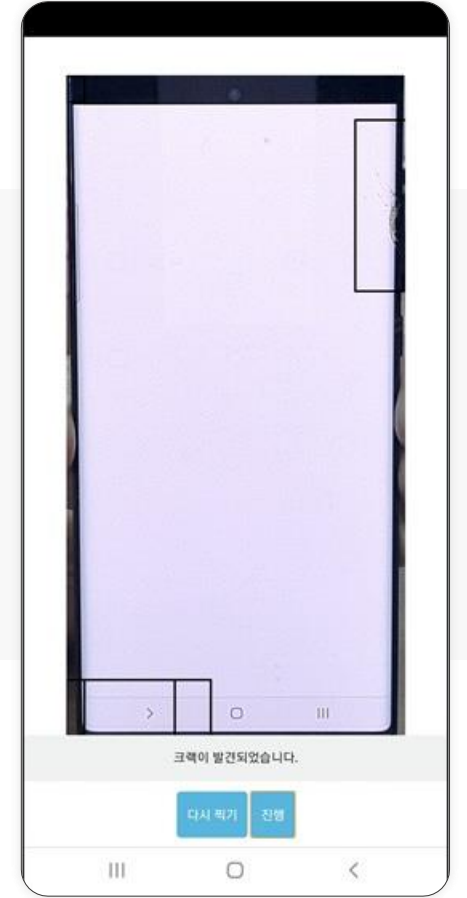
2. 객체촬영



3. 객체인식



4. Cropping

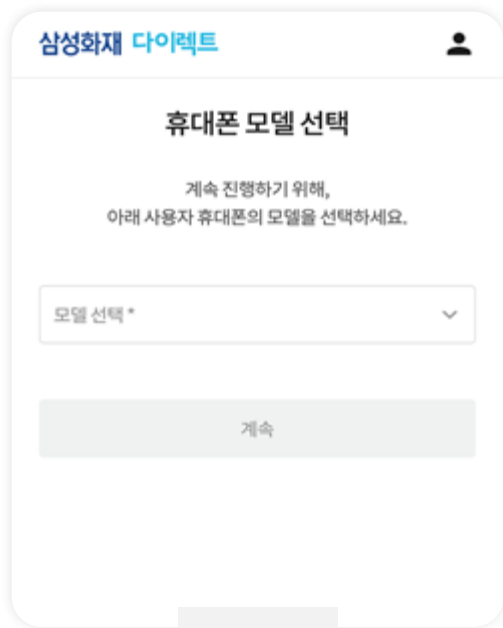


5. 결과값 출력

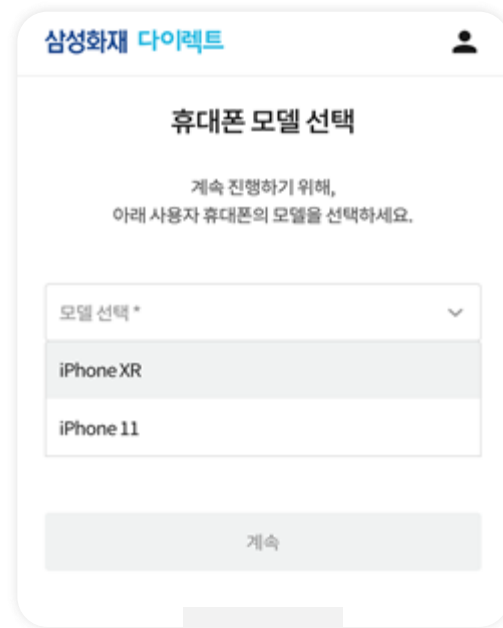
1단계

보험가입 가능한 폰 모델 구성하여 폰이 보험 가능한 폰인지 아닌지 시스템에서 걸러냄 (약 20개 정도의 모델 구성)

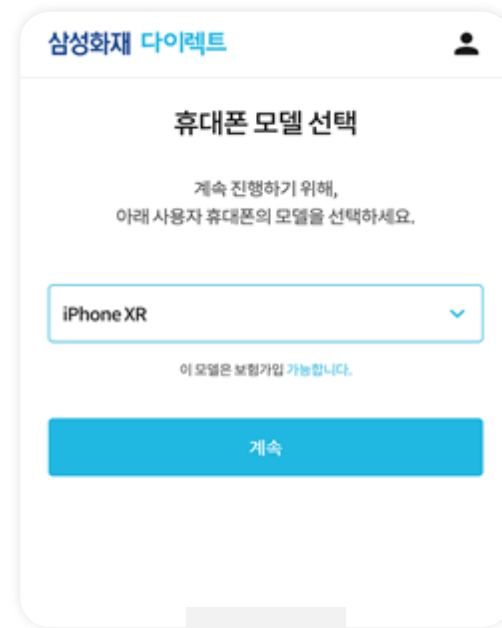
- 2-1. 폰이 보험가입 불가능 모델일 경우: "이 모델은 보험가입을 할 수 없습니다." 메시지 출현
- 2-2. 폰이 보험가입 가능 모델일 경우: "이 모델은 보험가입 가능합니다." 메시지 출현 후 보험 가입 진행하여 잼폰 솔루션으로 다음 단계 진행



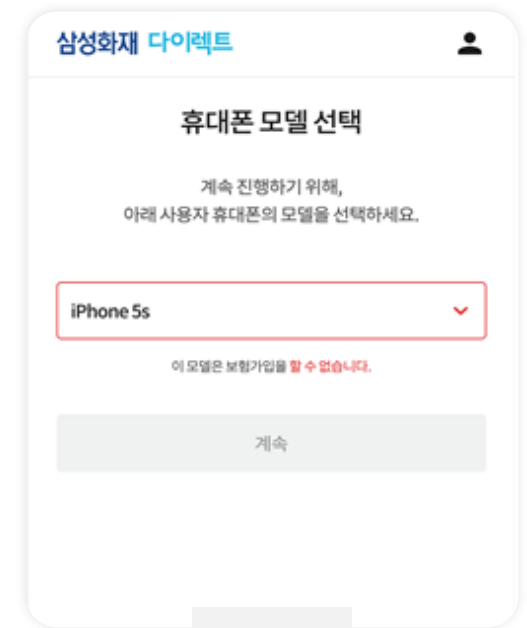
1-1



1-2



2-1



2-2

# 1) UI.UX 기획 및 디자인(5월4주차~6월3주차)

# 2. POC 결과 보고



## 2단계

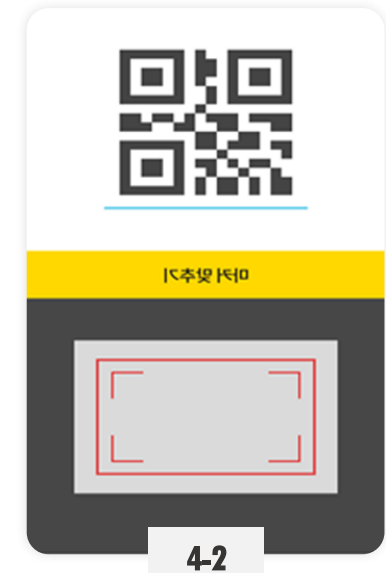
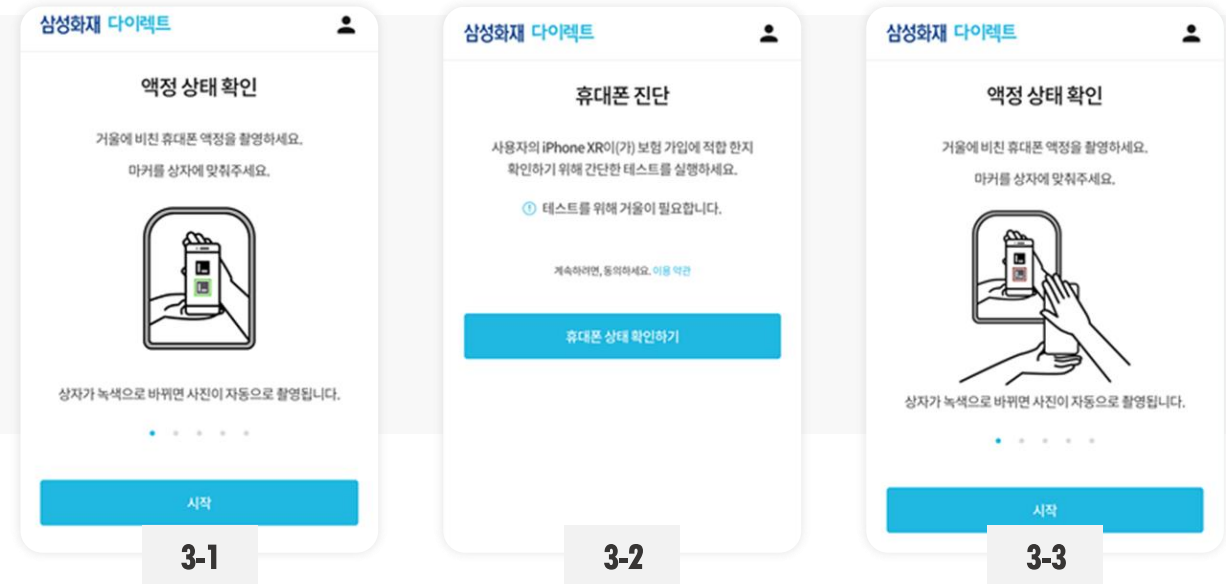
### 보험가입을 위한 촬영가이드

- QR 코드 사용안내 (촬영시 포커싱에 대한 안내)
- 촬영방법 등

## 3단계

### 기능개발

- 포커싱에 최적화된 방법에 대한 논의  
QR코드 도입 (도입시에 가장 쉽게 촬영가능한 디자인에 대한 고민)
- 해상도를 높이기 위하여 폰 밝기 조절 논의  
(배경색을 하얀색보다 약간 어두운 색으로 시도해볼 예정)
- 폰 네비게이션바 등 노이즈를 최대한 줄여서 촬영
- 촬영시 폰 카메라 리소스를 줄이는 방법에 대한 논의
- 촬영 후 데이터 처리 시간에 지루하지 않게끔 할 수 있는 방안 논의  
"처리중입니다." or 삼성화재 모델인 유재석 멘트 등



4단계

보험가입에 대한 가이드

- 3단계의 답변이 보험가입 불가능일 경우 : "액정 파손이 인지되었습니다. 한번 더 촬영을 원하시나요?" 메시지 출현
- 3단계의 답변이 보험가입 가능일 경우 : "축하합니다. 보험가입을 위한 다음 단계를 진행하겠습니다." 후 삼성화재 보험가입 절차를 밟게 됨





### Front-End 개발

1. 촬영: 3단계 개발
2. 촬영 시 Focusing 정확성 제고 - 해상도 Up, 노이즈 Down, 최대한 Fit하게 자르기
3. 안드로이드, IOS에서도 구동이 매우 잘되게 개발

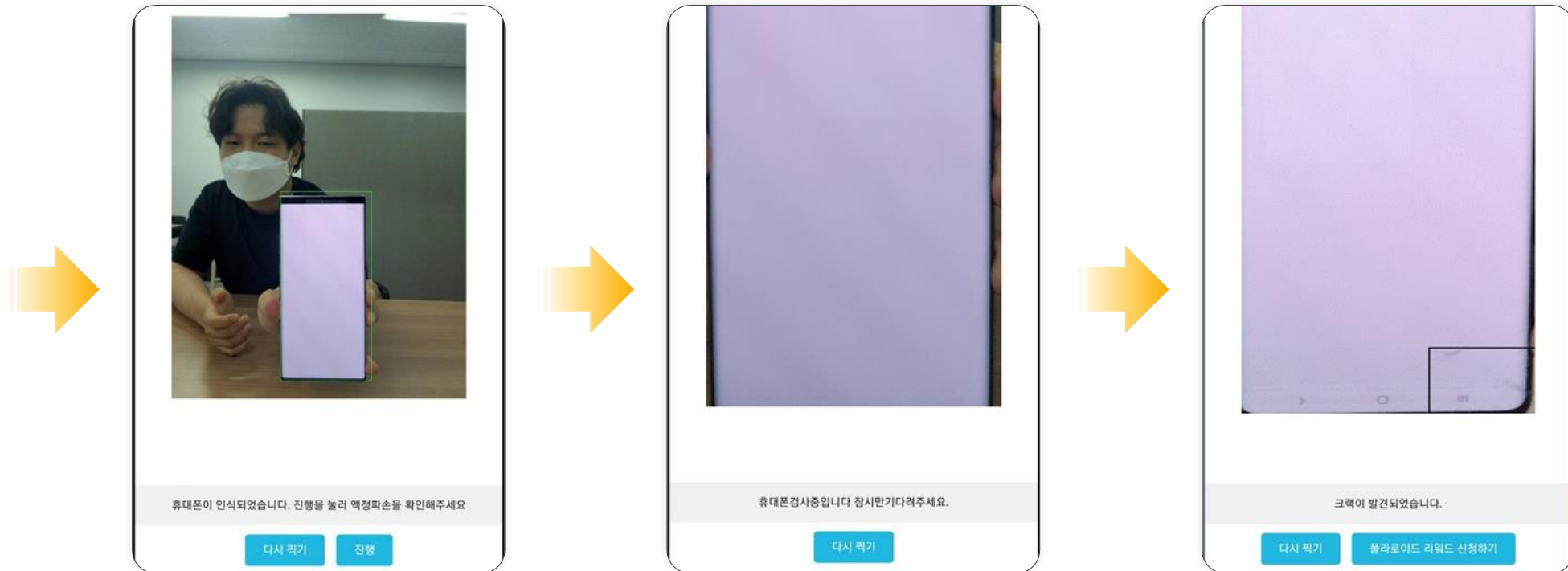
- 1단계: 포커스용 QR Code
- 2단계: 찍은 사람과 폰의 사진 출력
- 3단계: 필터링 기술을 이용하여 폰 사진만 Fit하게 출력



### Front-End 개발

1. 촬영: 3단계 개발
2. 촬영 시 Focusing 정확성 제고 - 해상도 Up, 노이즈 Down, 최대한 Fit하게 자르기
3. 안드로이드, IOS에서도 구동이 매우 잘되게 개발

- 1단계: 포커스용 QR Code
- 2단계: 찍은 사람과 폰의 사진 출력
- 3단계: 필터링 기술을 이용하여 폰 사진만 Fit하게 출력



## 실행

### 학습용 DATA 수집/확보(8,000여장)

- 당사에서 사업진행중인 Off-line 직매입을 통한 DATA수집
- 중고스마트폰 수출전문업체와의 제휴하여 개진폰 확보 및 사진촬영



## 추가 계획

- 당사에서 운영중인 AI 중고스마트폰 매입 사업을 통한 DATA수집
- 중고스마트폰 경매사이트 활용
- Pollide App을 통한 스마트폰 DATA수집

url	rec result	datetime
https://ai.jamfon.co.kr/images/1631259970534.png	[[{"Cracked": "[176, 862, 95, 901, 0.6666104], [57, 881, 95, 920, 0.62304705]"}]]	2021-09-10 07:46:10
https://ai.jamfon.co.kr/images/1631259998273.png	[[{"Cracked": ""}]]	2021-09-10 07:46:39
https://ai.jamfon.co.kr/images/1631260021298.png	[[{"Cracked": ""}]]	2021-09-10 07:47:01
https://ai.jamfon.co.kr/images/1631260182489.png	[[{"Cracked": ""}]]	2021-09-10 07:47:42
https://ai.jamfon.co.kr/images/1631260211721.png	[[{"Cracked": ""}]]	2021-09-10 07:50:11
https://ai.jamfon.co.kr/images/1631260247084.png	[[{"Cracked": ""}]]	2021-09-10 07:50:47
https://ai.jamfon.co.kr/images/163126037381180.png	[[{"Cracked": "[1291, 14, 378, 28, 0.82011802], [20, 1193, 38, 1237, 0.61773883]"}]]	2021-09-10 08:06:37
https://ai.jamfon.co.kr/images/163149664377.png	[[{"Cracked": "[1158, 334, 200, 378, 0.989744], [178, 334, 222, 378, 0.97633183], [334, 1404, 378, 1444, 0.989744], [178, 334, 222, 378, 0.97633183]"}]]	2021-09-10 01:34:24
https://ai.jamfon.co.kr/images/1631510146777.png	[[{"Cracked": ""}]]	2021-09-13 05:15:46
https://ai.jamfon.co.kr/images/1631510189006.png	[[{"Cracked": ""}]]	2021-09-13 05:16:35
https://ai.jamfon.co.kr/images/1631510329043.png	[[{"Cracked": "[1421, 40, 841, 80, 0.99998404], [220, 40, 360, 80, 0.99979321], [180, 40, 300, 80, 0.9528]"}]]	2021-09-13 05:18:45
https://ai.jamfon.co.kr/images/1631510439862.png	[[{"Cracked": "[1462, 22, 572, 44, 0.99998404]"}]]	2021-09-13 05:20:59
https://ai.jamfon.co.kr/images/16315105005172.png	[[{"Cracked": ""}]]	2021-09-13 05:21:45
https://ai.jamfon.co.kr/images/1631510909798.png	[[{"Cracked": ""}]]	2021-09-13 05:26:30
https://ai.jamfon.co.kr/images/1631511052356.png	[[{"Cracked": ""}]]	2021-09-13 05:30:52
https://ai.jamfon.co.kr/images/1631511581757.png	[[{"Cracked": ""}]]	2021-09-13 05:36:41
https://ai.jamfon.co.kr/images/1631511840022.png	[[{"Cracked": ""}]]	2021-09-13 05:45:40
https://ai.jamfon.co.kr/images/16315120076143.png	[[{"Cracked": ""}]]	2021-09-13 05:46:47
https://ai.jamfon.co.kr/images/1631512010078.png	[[{"Cracked": ""}]]	2021-09-13 05:46:50
https://ai.jamfon.co.kr/images/16315120716143.png	[[{"Cracked": ""}]]	2021-09-13 05:47:59
https://ai.jamfon.co.kr/images/1631512110281.png	[[{"Cracked": ""}]]	2021-09-13 05:48:30
https://ai.jamfon.co.kr/images/16315122082143.png	[[{"Cracked": ""}]]	2021-09-13 05:51:28
https://ai.jamfon.co.kr/images/1631512466697.png	[[{"Cracked": ""}]]	2021-09-13 05:54:26
https://ai.jamfon.co.kr/images/1631516826098.png	[[{"Cracked": "[1478, 1495, 607, 1530, 0.99976841]"}]]	2021-09-13 07:02:00
https://ai.jamfon.co.kr/images/1631516562048.png	[[{"Cracked": "[1110, 1530, 288, 1597, 0.99965121], [932, 1530, 421, 1575, 0.9891487], [510, 1508, 665, 2021-09-13 07:02:42]"}]]	2021-09-13 07:02:42
https://ai.jamfon.co.kr/images/1631516604084.png	[[{"Cracked": "[181, 1483, 142, 1503, 0.9960748], [284, 1483, 345, 1503, 0.98981548], [487, 1442, 528, 2021-09-13 07:03:24]"}]]	2021-09-13 07:03:24
https://ai.jamfon.co.kr/images/1631516649774.png	[[{"Cracked": "[1866, 1469, 628, 1511, 0.9876767], [524, 1469, 608, 1511, 0.731058002]"}]]	2021-09-13 07:04:09
https://ai.jamfon.co.kr/images/163151674784.png	[[{"Cracked": ""}]]	2021-09-13 07:05:47

8,000여장의 데이터 확보

깨짐,실금 등 다양한 깨짐 유형에 대해 약 10만 건 이상 (깨짐 객체 : 65,161건, 실금객체 : 38,161건) 의 학습셋을 구축하여 이 중 깨짐 객체만을 학습하여 판단 정확성 제고

1. 각 디바이스 혹은 해상도 별 임계치 조정

2. 거울 활용한 학습 셋 추가 확보 (실사용 환경 셋) 및 하스

3. Front-end 최적화 (촬영 속도 및 해상도, 객체 crop)

4. Back-end 및 객체 판별 모델 부하 테스트

5. 기존 시스템 전산망 연계 작업 (API 커스텀)

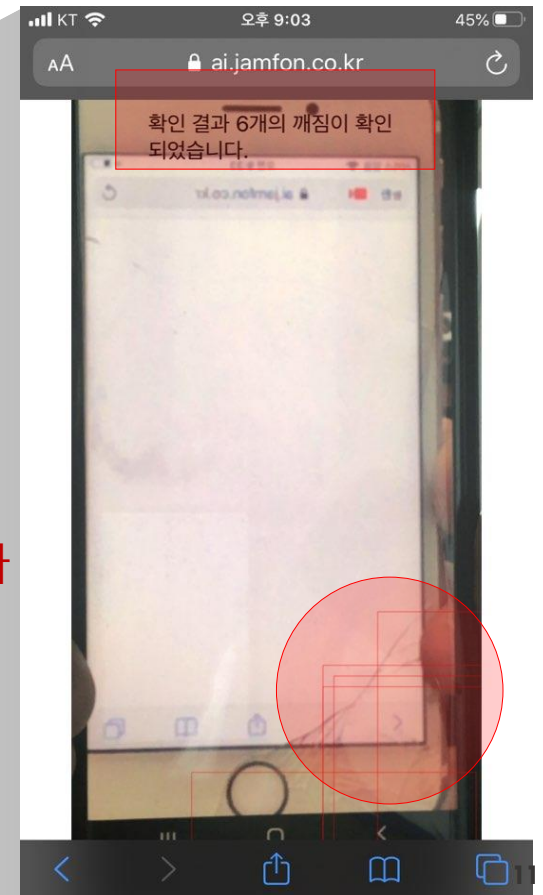
6. 모델 하이퍼파라미터 튜닝

[x1, y1, x2, y2 좌표, 예측치]

- [672, 1652, 868, 1736, 0.92459404],
- [784, 1484, 896, 1652, 0.9005258],
- [644, 1624, 812, 1764, 0.8925972],
- [896, 0, 1008, 84, 0.8710589],
- [644, 1680, 840, 1792, 0.8692436],
- [308, 1904, 420, 2044, 0.8534192]



임계치 80% 이상 깨짐에 대한 검출 결과



신규 라벨링 및 전이 학습을 통한 모델 개선

바운딩 박스

- 깨짐 객체 : 65,161개, 실금 객체 : 38,161개
- 이 중 깨짐 객체만을 학습하여 판단 정확성을 높임

이미지 수량

- 촬영된 이미지 총 7,600여 장 중  
노이즈 제거된 4,664장 활용

모델 학습 성능

- Epoch - 1,000회
- 활용 모델 - Faster R-CNN
- 학습, 검증 셋 비율 - 8 (3,731장) : 2 (933장)
- 모델 검증 Accuracy - 98.42%



<이미지 라벨링 및 모델 학습 개요도>

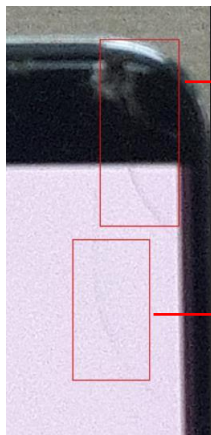
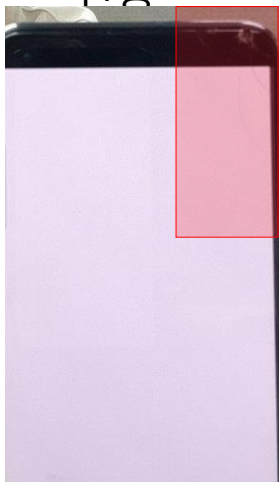
Faster R-CNN 모델의 특성을 활용한 깨짐 객체 분류 가능성 확인

모델 정확도

- Classifier accuracy for bounding boxed from RPN : 0.9842
- Loss Detector classifier : 0.050
- Loss Detector regression : 0.072

클래스 구분

- 실금의 경우 깨진 객체와의 특성 차이가 크지 않으며, 유사도가 임계가 비슷하여 깨짐 객체에 대한 정확도를 향상하는 전략 진행



→ 모델이 판단하기에 99.99% 수준으로 깨짐 객체로 판별

→ 모델이 판단하기에 78.93% 수준으로 깨짐 객체로 판별 (약간의 실금 가능성)

- 임계 수준을 향후 테스트 수준을 통해 설정

현재 기준 웨이브 99% 정도는 설정. 신규 케이스 때문에 더 낮게 설정 필요

모델 판단 결과 [{"Cracked": "[[620, 38, 697, 135, 0.99999547], [600, 290, 639, 348, 0.7893575]]"}]

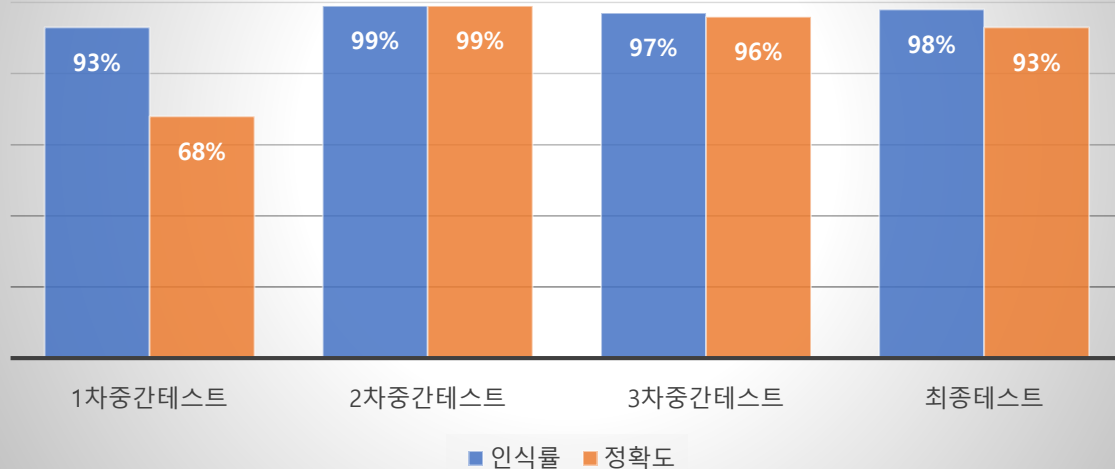
# 1) TEST 결과 및 추이 분석

# 3. TEST 결과 보고

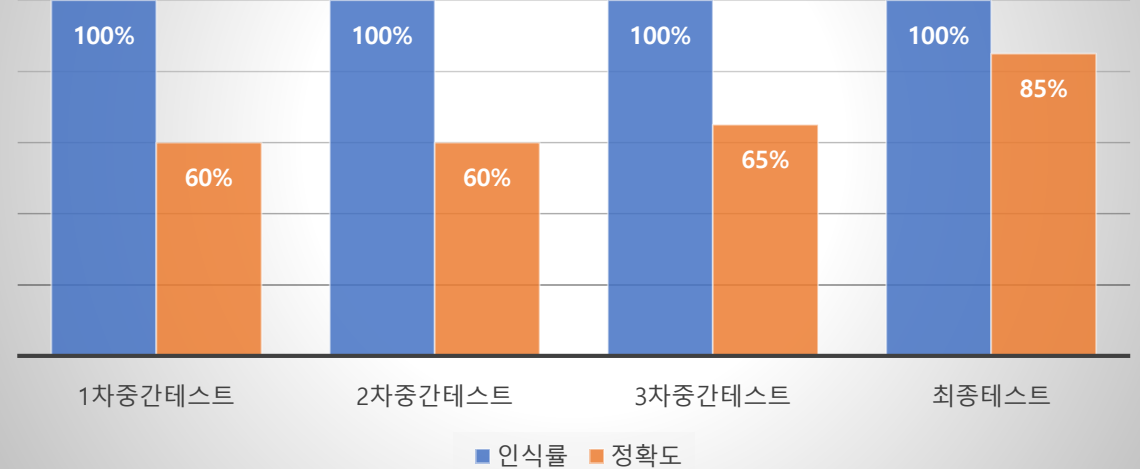


구 분	인식률						정확도			평균 시간	
	휴대폰전체			배젤안쪽			성공횟수	시도횟수	인식률		
	성공횟수	시도횟수	인식률	성공횟수	시도횟수	인식률					
파손	1차중간TEST	3.3	10	33%	9.3	10	93%	6.8	10	68%	11.87
	2차중간TEST	8.7	10	87%	9.9	10	99%	9.9	10	99%	
	3차중간TEST	6.8	10	68%	9.7	10	97%	9.6	10	96%	
	최종TEST	6.5	10	65%	9.8	10	98%	9.3	10	93%	
정상	1차중간TEST	1	10	10%	10	10	100%	6	10	60%	10.15
	2차중간TEST	9	10	90%	10	10	100%	6	10	60%	
	3차중간TEST	5.5	10	55%	10	10	100%	6.5	10	65%	
	최종TEST	6.8	10	68%	10	10	100%	8.5	10	85%	

### 파손폰 파손인식



### 정상폰 정상인식



API 구축	모델-시스템 연계	자체 개발하는 웹 시스템과 인공지능 모델 간의 연동 (응답 속도 5초 미만)
	모델-API 설계	타 보험사 간 연계 가능한 API System 설계
기술 상용화	특허출원	본 BM과 관련한 특허 출원
	계약체결	스마트폰 보험 관련 손해 보험사와의 계약 체결



### 모델-시스템 연계

- 휴대폰 객체를 벗어난 이미지 (손, 주변 사물 등)이 보이는 경우 깨짐 객체 판단에 불안정한 요소를 줄 수 있어, 휴대폰 객체 자체에 대한 학습 필요 혹은 현재 웹(CAFE24)에서 휴대폰 객체를 판단하는 COCO-SSD 모델의 대체 방안 강구
- 타이트하게 촬영 가능하도록 현재 촬영되는 이미지의 마진 비율 조정 논의

### Moral Hazard

- QR 코드의 촬영 전 인식 여부 확인 (웹 개발)
- 1회의 촬영이 아닌 2~3회 정도의 촬영을 요청
- 문제가 있을 것으로 의심되는 단말 (어둡거나 일부러 떨림 등)에 대해서는 모니터링 후 검수 필요
- 일부러 단말을 흔드는 현상은 앱인 경우 자이로, 가속도 센서 이용하여 재 촬영 요구 가능

### Infra

- 서버 환경 셋팅 및 분산처리 환경 구축(LB, VPC, SUBNET 등)
- 모델 코드 효율적인 연산 가능하도록 리팩토링
- 서버 보안, 이중화 작업, DB작업 필요

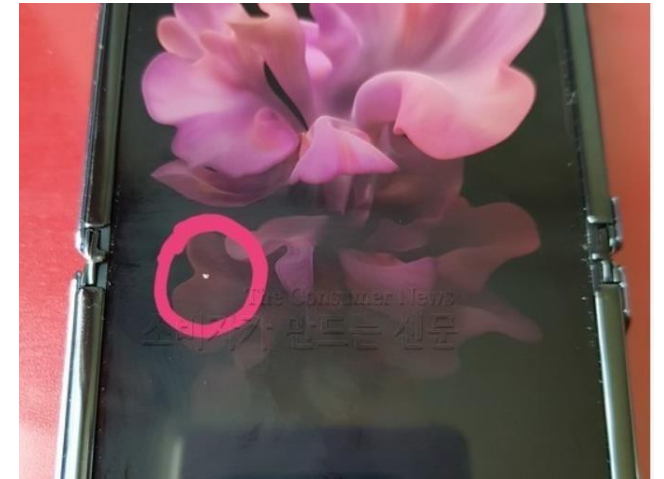
#### CASE

- 먹물 현상 (떨어뜨리거나 사제 액정 필름)
- 일자로 접힌 부분이 흰색으로 뜨는 현상
- 힌지 주변 주름이 생기는 현상
- 뒷 부분의 깨짐 현상
- 접히는 부분에 흰 점이 생기는 현상



#### 처리방안

- GAN 알고리즘을 통한 케이스 증대 (실제 데이터보단 정확도 낮음)
- 별도의 모델이 필요할 가능성이 있음



# Thanks!

주식회사 맛앤멋



```
[{"Cracked": "[[585, 22, 697, 157, 0.99912745], [22, 247, 67, 292, 0.99504143], [360, 22, 450, 90, 0.99464816], [517, 270, 630, 315, 0.993079], [180, 382, 247, 427, 0.8871206], [697, 315, 832, 360, 0.66159016], [405, 67, 540, 202, 0.6166686], [427, 45, 585, 157, 0.5853242], [292, 292, 472, 382, 0.568878]]"}]
```

