서론

자동화 펀칭 머신을 도입하여 기업 “포스-벽진”은 야간작업 진행을 계획했다. 하지만 무인환경에서 설비의 금형 파손 시 대처 시스템의 부재로 불량품 발생 및 재고자산손실 등의 문제를 겪고 있었다. 야간 생산은 월간 약 15일이며 월간 평균 불량률은 7% 이다. 하지만 인력 부족으로 인해 야간 근무자 배치가 어려워 대처방안으로 설비의 운영 속도를 70% 수준으로 제한하고 야간 생산량을 감소시켜 손실비용을 감소시켰다. 기존 대처방안의 손실비용은 월 평균 불량품 85개 라고 가정하고, 철판손실비용, 폐기처리비용, 주간 추가작업비용을 계산한 결과 약 4,590,000원이다. 만약 금형파손 발생 1시간 뒤 작업자가 조치를 취한 경우를 가정하여 손실비용을 계산한 결과 1,171,500원으로3,418,500원을 절약한다. 본 연구는 펀칭 설비의 음향을 인식하고 낙하 하는 스크랩을 인식하여, 두 데이터를 바탕으로 종합적인 의사결정을 통해 설비 금형의 파손 여부를 판단하는 문제를 다룬다. 또한, 수집된 금형의 펀칭 횟수와 펀칭 대상 철판의 두께 데이터를 바탕으로 최적의 금형을 추천하고 파손 시기를 예측하는 프로세스를 다룬다. 음향 인식 기법으로는 General Adversarial Networks 의 Discriminator 를 활용하여 정상과 비정상 음향을 구분 하도록 한다. 펀칭 설비 하단에서 낙하 하는 스크랩은 적외선 센서를 활용하여 인식한다. 그리고 음향 데이터와 스크랩 데이터는 실시간으로 종합적인 의사결정을 내리는 프로세스에 사용된다. 주간 혹은 야간의 작업 환경에 맞게 최적의 금형을 추천하고 파손 시기를 예측하는 프로세스는 어플리케이션을 통해 구현된다. 작업자가 작업 설계도를 바탕으로 설비의 금형 펀칭 횟수를 입력하면, 추천 프로세스에 맞춰 작업자에게 해당 금형을 추천하고, 파손이 예측될 시에는 교체 알림을 하여 파손으로 인한 재고자산손실과 불량품 발생을 최소화 한다.

Key words : Generative Adversarial Networks, Discriminator, SpecGAN, RaspberryPi, Arduino

**목 차**

**1. 프로젝트 요약문**

**2. 프로젝트 소개**

2.1프로젝트 주제

2.2 프로젝트 필요성

**3. 프로젝트 내용**

3.1 Error Prevention System(EC System)

음향 인식

- 데이터 수집

- 데이터 전처리

- GAN(Generative Adversarial Networks)

- 교차검증(K-fold cross validation)

- 실시간 데이터 처리

적외선 인식

의사결정

3.2 Error Catcher System

누적 펀칭 횟수 수리모형

**프로젝트 결과물**

**기대 효과**

**참고문헌**

**별첨**

**표 목차**

<Table 1. 프로젝트 요약>

<Table 2. 대안 비교>

<Table 3. 금형 종류>

<Table 4. 교차검증>

<Table 5. 종이 실험>

<Table 6. 최종 결과물 활용 계획>

<Table 7. 금형 모양>

<Table 8. 철판 종류 표>

**그림 목차**

<Picture 1. 프로젝트 소개>

<Picture 2. 현장라인 데이터 수집 모습>

<Picture 3. VNC>

<Picture 4. 데이터 전처리>

<Picture 5. 컨베이어 벨트>

<Picture 6. 보조기구>

< Picture 7. 보조기구 설계도>

<Picture 8. 1일 동안 사용>

<Picture 9. 2일 동안 사용>

< Picture 10. 종이 보관함>

<Picture 11. 의사결정 로직>

< Picture 12. 의사결정>

< Picture 13. A\_RO 예시>

<Picture 14. 정상 환경일 때>

<Picture 15. 비정상 환경일 때>

<Picture 16. 메뉴 구조도>

<Picture 17. 앱 메인 화면>

<Picture 18. 금형 추천(주간)>

<Picture 19. 현재 사용 중 금형>

<Picture 20. 생산 정보 입력>

<Picture 21. 생산 정보 확인>

<Picture 22. 설계도 목록>

<Picture 23. 설계도 등록>

<Picture 24. 금형 사용현황 목록>

<Picture 25. 금형 사용현황 세부>

<Picture 26. 금형 교체 경고>

<Picture 27. 금형 파손 히스토리>

<Picture 28. 금형 파손 등록>

<Picture 29. 회사 조감도>

<Picture 30. 설비 종류>

<Picture 31. 주요 제품 >

<Picture 32. 금형>

<Picture 33. GAN>

<Picture 34. 시스템 ERD>

<Picture 35. Hidden layer 구조도>

1. **프로젝트 요약문**

|  |  |
| --- | --- |
| 프로젝트 주제 | (주) 포스-벽진 자동화 설비 개선 |
| 프로젝트 배경 | 자동화 설비를 도입한 포스-벽진  무인환경에서 금형 파손 시 대처 시스템 부재  실시간으로 금형 파손을 인식할 수 있는 시스템이 요구  금형 파손을 예측할 수 있는 시스템이 요구 |
| 프로젝트 목표 | 금형 파손 시 작업자에게 알림 시스템 구현  금형 추천 시스템 및 PUSH 알림App구축 |
| 프로젝트 내용 | 불량 인식 시스템   1. 음향 인식   금형 펀칭 음향 수집 및 전처리  GAN 학습  신뢰성 검증  펀칭음향 정상 / 비정상 구별   1. 적외선 인식   Arduino와 적외선 빔 센서를 통해 scrap 인식  블루투스 모듈을 통해 인식 결과 값 App에 전송  설비 가동상태 인식  금형 추천 App 구축  설계도 데이터 수집 및 입력  금형 추천 시스템 프로세스 정립  파손 예측 알고리즘 구축  어플리케이션 개발 |
| 프로젝트  최종 결과물  활용계획 | 2학기 이공계 학술제 출품  교외 대한산업공학회 공모전 출마  (주)포스-벽진 사에 시스템 개선 및 App 구축 |

<Table 1. 프로젝트 요약>

1. **프로젝트 소개**

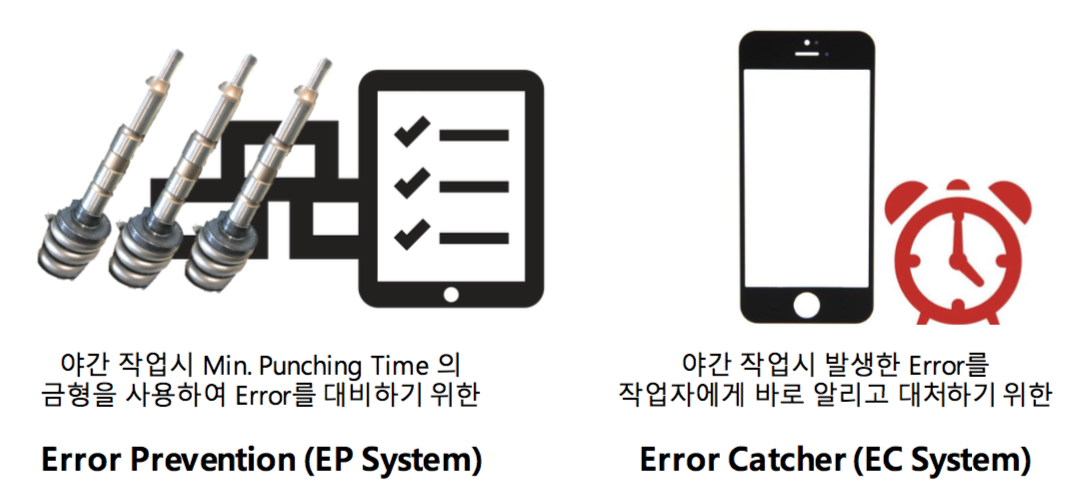
- 프로젝트 주제 : 딥러닝을 활용한 자동화 설비 개선

1. Error Catcher (EC) System구축

야간 작업 시 발생한 Error를 작업자에게 바로 알리고 대처하기 위한 시스템. 철판 scrap이 떨어지는 여부를 판단하는 적외선 인식기와 punching소리로 비정상과 정상 소리를 판단하는 음성인식을 종합하여 에러를 판단하고, 에러 발생 시 App을 통해 알람을 보낸다.

1. Error Prevention (EP) System 구축

주간&야간 작업 시 추천 로직을 통해 선택된 금형을 사용하여 Error를 대비하기 위한 금형 추천 시스템. 총 26가지의 금형을 각각 3개씩 유지하여 생산할 때 금형을 추천한다.



<Picture 1. 프로젝트 소개>

- 프로젝트 필요성

현재 야간 생산량 = 15days / month

평균 불량 횟수 = 2.5 days / month

금형 종류 = 총26개

금형 비용 = 2,284,000 (￦) (26개의 금형의 총합)

예방 관련하여 모든 금형을 교체하는 것을 고려했을 때, 나오는 경우의 수와 각 대안의 비교는 다음과 같다.

|  |  |  |
| --- | --- | --- |
| **경우의 수** | **비용** | **비교 분석** |
| **A**. 1일에 한번씩 금형 교체 | 고정비(3426만원) +  금형재고비용 (악성 재고) | 에러가 나올 확률이 없지만, 막대한 재고 비용과 복잡한 재고관리가 요구된다. |
| **B**. 3일에 한번씩 금형 교체 | 고정비(1142만원) +  금형재고비용(악성 재고) | A의 대안보다는 비용이 적지만 여전히 높은 재고 비용과 복잡한 재고관리가 요구된다. |
| **C**. 5일에 한번씩 금형 교체 | 고정비(685만원) +  금형재고비용(악성 재고) | **회사에서 해본 결과 재고관리와 재고비용적인 측면에서 부담을 느껴서 중단하였다.** |
| **D**. 15일에 한번씩 금형 교체 | 고정비(228만원) +  금형재고비용(악성 재고) | 한달 오류 횟수를 충족하지 않은 경우이므로 고려하지 않는다. |
| **E. 금형 추천 , error 발생 후 알림** | **1,171,500 원의 비용이**  **발생** | 금형 추천을 통해 error 발생 확률을 최대한 낮추고 알림을 통해 에러를 대처할 수 있게 때문에 가장 합리적인 대안이다. |

<Table 2. 대안 비교>

C 대안을 포스 회사가 직접 적용 해 보았지만, 재고관리와 재고비용적인 측면에서 부담을 느꼈다. 따라서 금형 추천을 통해 에러를 대비하고, 더 나아가 차후 Error 상황을 대처하기 위한 알림 시스템이 필요하다.

1. **프로젝트 내용**
   1. Error Catcher System(EC System)

음향인식

본 장에서는 General Adversarial Networks 의 학습 데이터인 펀칭 음향은 금형의 종류에 따라 달라지게 된다. 금형은 크게 크기에 따라 A, B, C, D, E, G 총 6개의 종류로 이루어져 있다. 또한 각 금형의 종류에서 세부적으로 펀치 바디(punch body)의 모양에 따라 종류가 나뉘어 26가지의 금형이 존재한다. 본 연구에서는 같은 종류의 금형은 같은 카테고리로 간주하여 6개의 카테고리의 음향 데이터 학습을 진행하였다.

|  |  |  |  |
| --- | --- | --- | --- |
| 금형 | 펀치바디 | | |
| A | SQ | 0B\_W | RE\_W |
| RO | OB\_H | RE\_H |
| CP | SP |  |
| B | OB\_H | RO | CR |
| RE\_W | SQ |  |
| RE\_H | SQ\_D |  |
| C | OB\_W | OB\_H | CR |
| SP | RR |  |
| D | RE\_W | RE\_H |  |
| E | RE\_W | RE\_H |  |
| G | CR | RE\_W |  |

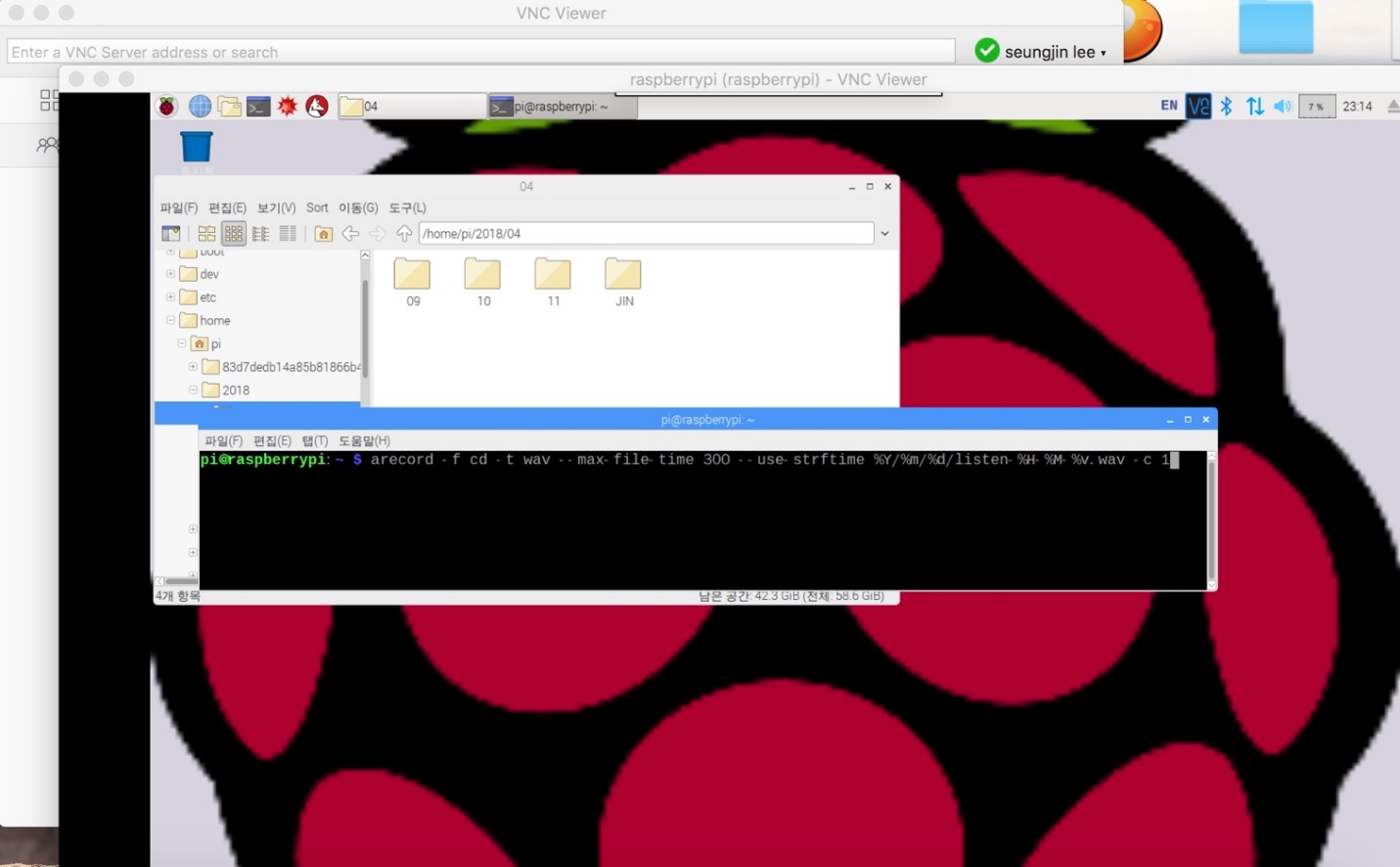
<Table 3. 금형 종류>

- 데이터 수집

포스-벽진의 생산 라인에 설치한 Raspberry Pi 3 model B에서 Python의 pyaudio 라이브러리를 사용하고 USB 마이크를 통해, 5분 주기로 수집을 실시하였다. 수집기간은 2018년 3월 16일부터 5월 11일 까지 이다. 펀칭머신의 작업 특성상 생산하는 제품의 형태에 따라 여러 종류들의 금형이 번갈아 가며 펀칭을 하여, 소리 데이터가 불규칙하게 수집된다. 수집된 음향 데이터는 금형의 종류별로 카테고리를 나누어 학습을 실시할 필요성이 있으므로, Audacity 음향 편집 프로그램을 사용하여 1차적으로 5분 단위로 수집된 음향 데이터를 같은 금형의 종류별로 구분하였다.



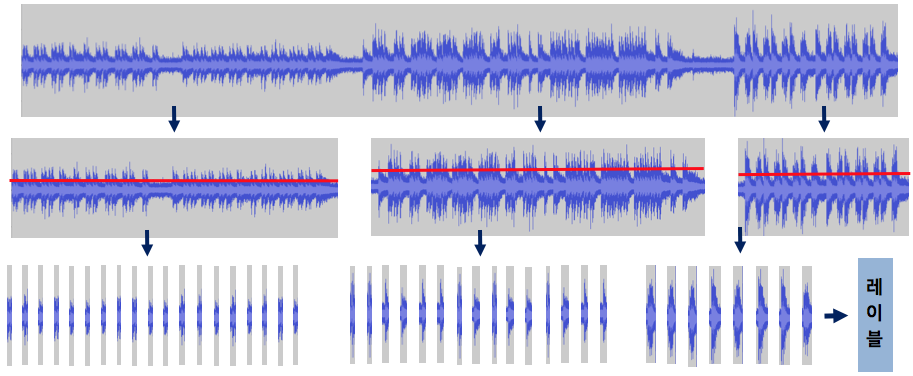
<Picture 2. 현장라인 데이터 수집 모습>



<Picture 3. VNC>

- 데이터 전처리

금형의 종류별로 구분하여 모은 음향 데이터들은 Python의 pydub 라이브러리를 통해 1회 펀칭 단위로 나누었다. 이 때 데시벨(decibel)이 나누는 기준이 되는데, 금형 마다 기준 데시벨이 다르므로, 각 금형에 적절한 데시벨로 자르게 된다.구현한 Python 소리 편집 프로그램은 특정 주파수 이상이면 편집되는 프로그램이다. 최종적으로 편집된 데이터는 금형 별로 labeling 되며 training 과정으로 넘어가게 된다.



<Picture 4. 데이터 전처리>

GANs는 이미지 합성 문제(the problem of synthesizing realistic images)에 적합하다고 알려져 있어 음향 합성 문제를 해결하기 위해서는 음향 데이터를 스펙트로그램 (spectrogram) 으로 변형하여 해결할 필요가 있다(Donahue, C., McAuley, J. & Puckette, M., 2018). 그러므로 GANs 학습에 앞서, 정리된 데이터들을 멜 스펙트로그램(Melspectrogram)으로 변환시키는 전처리 과정을 진행하였다. 멜 스펙트로그램은 짧은 시간의 퓨리에(Fourier)변환에 유용하게 쓰인다(K.Choi, G.Fazekas, K.Cho, and M.Sandler, 2017). 그 이유는 멜 스펙트로그램은 짧은 시간의 데이터를 변환함에도 불구하고 충분한 정보를 포함하고 있기 때문이다(Choi, K., Fazekas, G., Sandler, M. & Cho, K., 2018). 전처리 과정은 데이터들의 평균과 분산 값들을 구하여 정규화된 멜 스펙트로그램 이미지를 생성하게 된다. 그리고 이 모든 데이터들은 .npz 파일로 저장되게 된다. 상기 과정은 (Specgan-generate audio with adversarial training, 2018)을 참고했다.

- GAN(Generative Adversarial Networks)

GAN는 Discriminator 와 Generator로 구성이 된다. Discriminator는 입력되는 값이 실제값인지 여부를 구분하는 역할을 담당한다. 그리고 Generator는 임의의 값을 받아들여 실제값과 유사한 값을 도출하도록 노력한다. 예를들어 실제 값을 X, 임의의 값을 N라고 가정했을 때, Discriminator(X)=1, Discriminator(N)=0 이 되도록 노력한다. Generator(N) 는 X값이 되도록 노력한다.

결과적으로 Generator는 Discriminator(Generator(N))=1이 되도록 노력한다. 본 연구에서는 Generator가 생성한 실제값과 유사한 값은 사용하지 않고, Discriminator의 실제값과 유사값을 구분하는 부분만을 활용했다. 기존 드럼의 소리를 구분하는 SpecGAN연구 (Specgan-generate audio with adversarial training, 2018)를 참조하여 Discriminator에 정상의 실제 정상 펀칭음향을 학습시키고, 정상의 펀칭음향과 실제 비정상의 펀칭음향을 구분하도록 했다. Batch: 64, Epoch: 3000, hidden layer: 3으로 학습을 실시했으며, TanH와 ReLU 함수를 이용하여, Discriminator가 정상이라고 판단하면 표현할 수 있는 만큼의 가장 큰 숫자를 Output으로 나타내며, 비정상일 경우 가장 작은 숫자로 나타내도록 설계했다.

- 교차검증(K-fold cross validation)

본 연구에서 검증은 5 fold cross validation을 수행하였다. 폴드(fold)는 각각의 데이터 수에 따라 비례하여 20 %를 테스트 데이터로 사용하였으며, 각 폴드를 한 번씩의 테스트 셋(Test set)으로 하여 총 5회의 학습&테스트 과정을 반복하였다. 각 학습 반복 횟수는 3,000이며, 인식률은 다음과 같다.

|  |  |  |
| --- | --- | --- |
| 정확도 | 정상 음향 (%) | 비정상 음향 (%) |
| 폴드 1 | 94.83 | 91.91 |
| 폴드 2 | 93.20 | 91.50 |
| 폴드 3 | 97.33 | 93.32 |
| 폴드 4 | 94.61 | 92.12 |
| 폴드 5 | 95.35 | 92.43 |
| 평균 | 95.06 | 92.26 |
| 표준편차 | 1.50 | 0.68 |

<Table 4. 교차검증>

* 실시간 데이터 처리

앞서 SpecGAN 학습을 위한 데이터 수집 과정과 같은 방식으로 라즈베리파이 3 모델 B에서 Python의 pyaudio 라이브러리를 사용하여 3초 주기로 작동하고있는 펀칭머신의 천공 소리를 녹음한다. 다음으로 3초 동안 다수의 펀칭 음향이 녹음된 .wav파일을 자동으로 Python의 pydub 라이브러리를 사용하여 1회 펀칭 단위별로 음향 파일을 나눈다. 이렇게 나뉘어진 음향 파일들은 순차적으로 D을 통해 정상 펀칭 음향인지 비정상 펀칭 음향인지 판별된다. 앞서 SpecGAN 학습을 위한 데이터 정리 과정에서 1회 펀칭 단위별로 음향 파일을 나눌때에는, 금형의 종류별로 정리가 되어있는 상태에서 나누기 때문에 각 종류별로 데시벨의 적절한 기준을 정하여 나눌 수 있었다. 하지만 실시간 데이터 처리에서는 여러 종류의 금형의 소리가 섞여있는 음향 파일을 자동으로 1회 펀칭 단위로 나눌 필요가 있다. 방법은 다음과 같다. 데시벨의 기준을 크게 두가지로 나누어, 먼저 첫번째 데시벨의 기준으로 음향을 자르고 다음으로 두번째 데시벨의 기준으로 음향을 잘라 D를 통해 도출된 두 결과값 중에서 큰 값을 채택한다. 두 데시벨 중에서 더욱 적절한 데시벨 기준을 채택하게 되는 것이다. 이렇게 도출된 값들의 평균은 데이터베이스에 최종적으로 업로드(upload)된다. 데이터베이스는 MySQL을 사용한다.

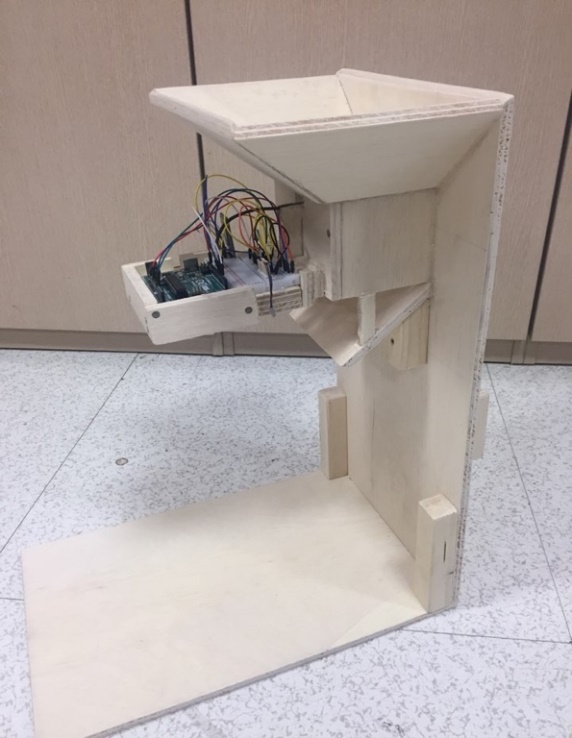
적외선 인식

본 장에서는 금형이 철판을 펀칭하고 난 후 펀칭머신 하단에서 낙하 하는 스크랩은 인식하는 과정을 다룬다. 이 방법은 만약 금형이 파손되어 실제 펀칭을 안했지만, Discriminator가 정상이라고 오판단 했을 경우를 보완할 수 있다. 스크랩은 실제 펀칭이 되었을 경우에만 낙하가 되므로 불량인식 정확도를 향상시킬 수 있는 방안이다. 스크랩은 폭 25cm인 컨베이어 벨트를 통해 낙하하는데, 이 부분에서 낙하하는 스크랩을 직접적으로 인식하기에는 낙하속도가 빠르고 스크랩의 크기가 매우 작아 한계점이 있다. 그래서 폭을 감축하고 낙하속도를 낮추는 보조기구를 제작했다.



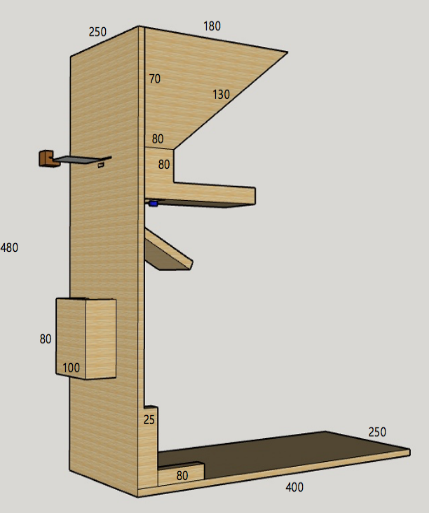
<Picture 5. 컨베이어 벨트>

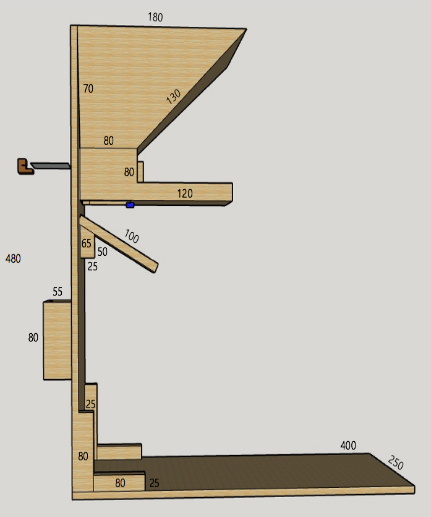
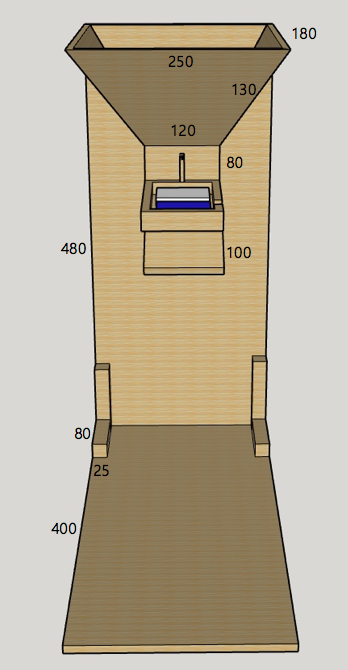
상단 스크랩이 낙하 하는 부분은 25cm에서 10cm로 좁아지는 형태를 하며, 10cm로 좁아진 부분에는 종이를 받쳐 간접적으로 스크랩을 인식하도록 했다. 낙하 하는 스크랩이 종이에 닿으면 종이가 구불어지게 되고, 적외선 센서는 이 종이의 흔들림은 인식하여 스크랩 낙하 여부를 인식한다. 이 부분에서 종이의 내구성을 최대로 하면서, 인식률이 가장 높은 조건을 실험을 통해 도출했다. 총 7개 조건의 종이를 바탕으로 8시간 실험을 실시한 결과, 7번 조건-종이 1장 + 받침대(OHP 1장)이 최적 조건임을 도출했다. 아두이노는 적외선센서를 통해 스크랩이 인식되면 1의 값을, 인식되지 않으면 0의 값을 출력하며 이 값은 앞서 처리한 음향인식 값과 같이 데이터베이스에 업로드된다.

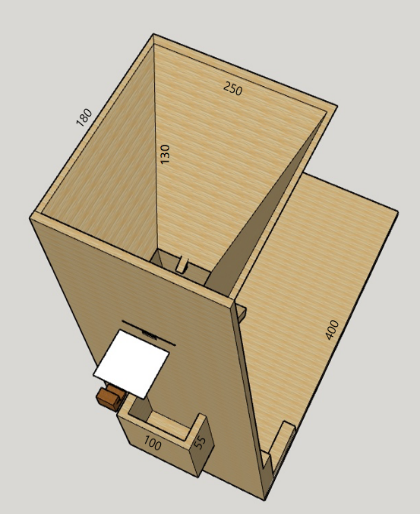
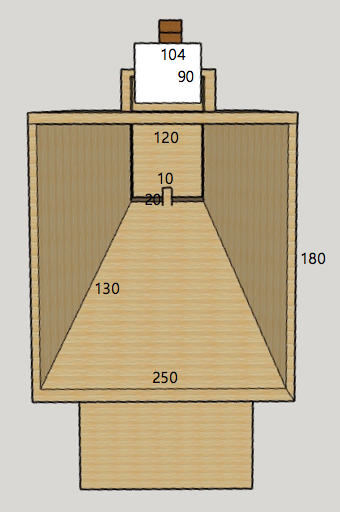


<Picture 6. 보조기구>

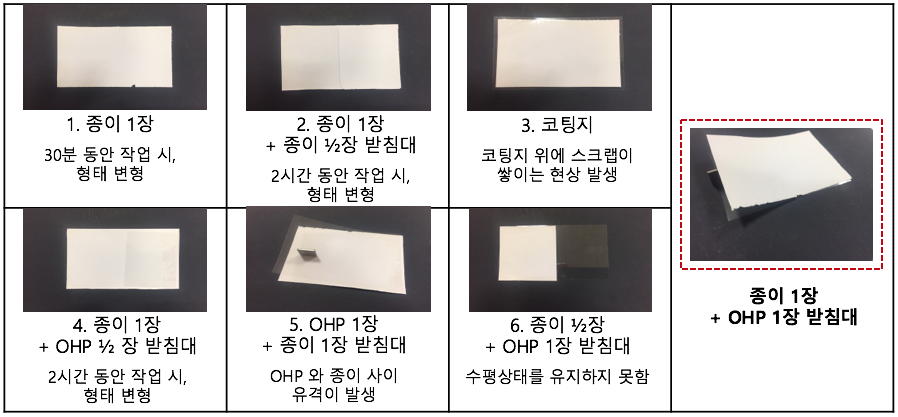
인식률을 최대한 높이기 위해 총 4차례에 걸쳐서 보조기구를 만들었다. 첫번째 보조기구는 scrap이 적외선 인식기를 거치지 않고 그냥 지나쳤다. 이 문제를 보완하기 위해 <Picture 7> 과 같이 적외선 인식기로 scrap이 떨어지도록 유도하는 덮개를 만들었다. 이 보조기구를 설치하기 위하여 현장에 방문하였으나 보조기구의 높이가 잘못 측정되어 있어서 설치가 불가 하였다. 그래서 다시 측정한 높이에 맞추어 보조기구를 수정하였고, 재방문하여 설치를 시도했다. 하지만 scrap이 적외선 인식기에 도달하지 못하는 결정적인 문제가 발생하였다. 보조기구의 설계에 문제가 있음을 인식하고 재설계를 하였으며, 최종적으로 설치를 마무리 하였다. 보조기구의 설계도는 다음과 같다.







< Picture 7. 보조기구 설계도>



<Table 5. 종이 실험>

적외선 인식기 사이의 종이의 종류를 선택하기 위해, 실험을 하였다.

1번 : 종이 1장을 사용한 결과, 30 받분 동안 scrap이 떨어졌을 때 형태 변형이

일어났다.

2번 : 종이 1장과 종이 반 장을 받침대로 사용한 결과, 2시간 동안 scrap이

떨어졌을 때 형태 변형이 일어났다.

3번 : 코팅지를 사용한 결과, 큰 scrap이 떨어지는 경우에는 잘 인식하지만, 크기가

작은 scrap들은 코팅지 위에 쌓이는 현상이 발생했다.

4번 : 종이 1장과 OHP필름 반 장을 받침대로 사용한 결과, 2시간 동안 scrap이

떨어졌을 때 형태 변형이 일어났다.

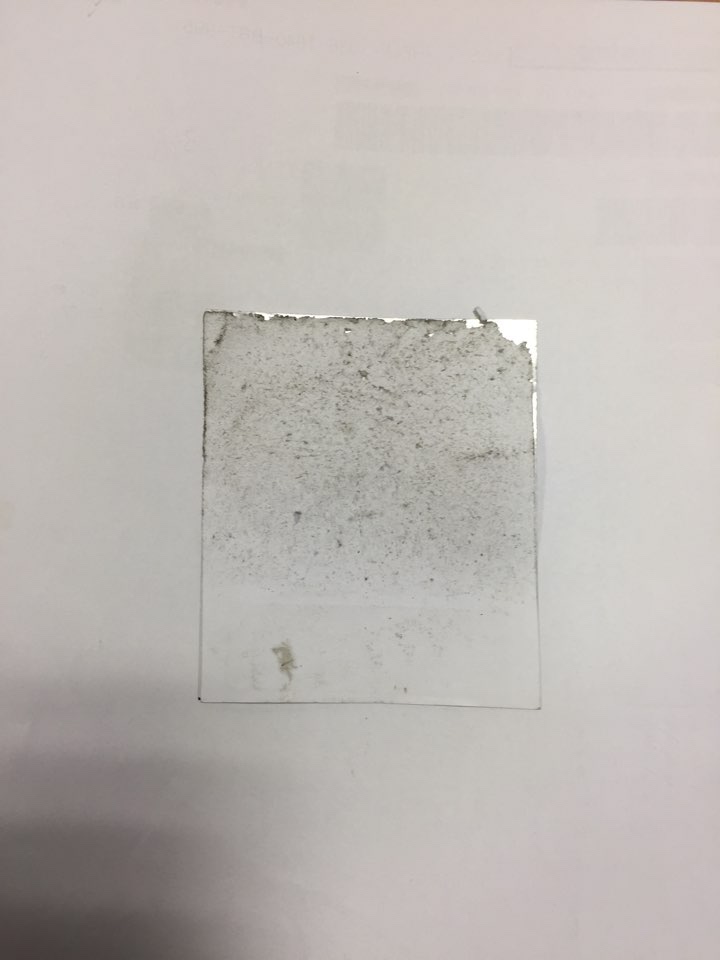
5번 : OHP 필름 1장과 종이 1장을 받침대로 사용한 결과, OHP 필름과 종이 사이

유격이 발생하였다.

6번 : 종이 반 장과 OHP 필름 1장을 받침대로 사용한 결과, 수평상태를 유지하지

못하는 현상이 발생했다.

종이 1장과 OHP 필름 1장이 받침대로 사용되었을 때 최적의 조건임을 발견하였고, 이 조건으로 실험 및 검증을 진행하였다.

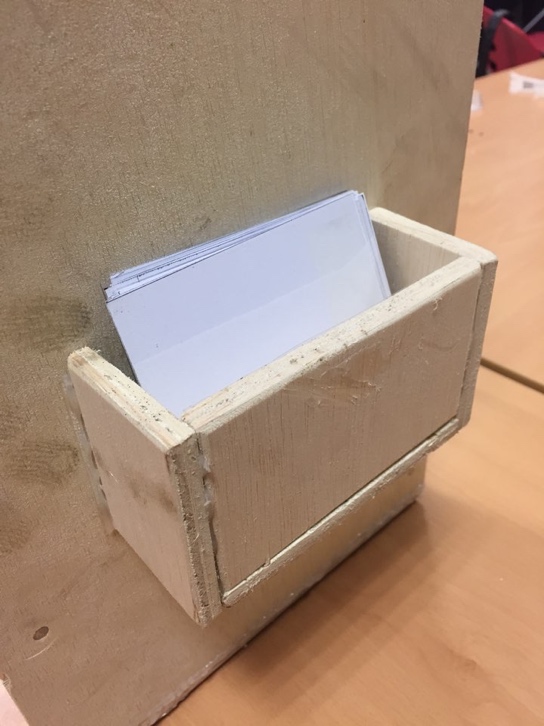
 

<Picture 8. 1일 동안 사용> < Picture 9. 2일 동안 사용>

위의 사진은 최적의 조건으로 선정된 종이1장과 OHP 필름 1장을 받침대로

사용하였을 때의 사진이다.

<Picture 8> 은 실제 현장에서 1일 동안 사용한 종이이고 <Picture 9> 은 2일 동안 사용한 종이이다. 3일을 사용하게 될 경우 종이의 인식 부분이 찢겨서 인식의 정확도가 낮아지는 현상이 발생하였다. 따라서 적외선 인식기 보조기구 뒤에 종이 보관함을 만들어 작업자가 편리하게 종이를 바꿀 수 있도록 설계하였다.

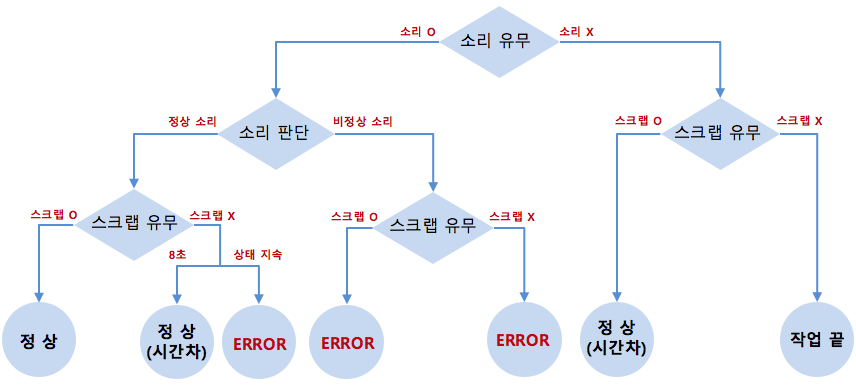


< Picture 10. 종이 보관함>

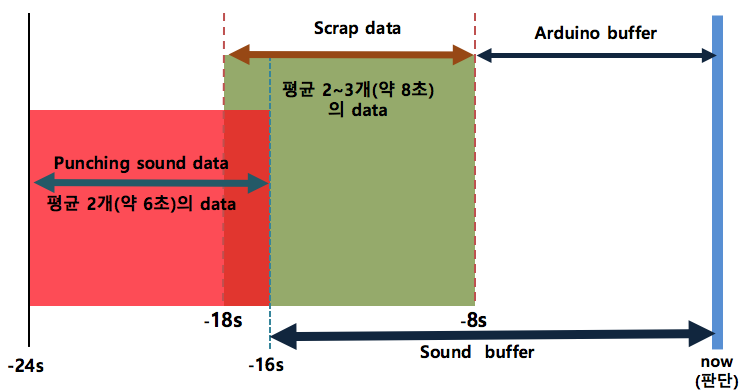
의사결정

본 장에서는 SpecGAN 학습으로 생성된 Discriminator의 값과 적외선센서를 통해 인식한 아두이노의 값을 종합하여 의사결정을 진행한다. 스크랩은 펀칭 소리가 난 뒤에 컨베이어 벨트를 따라 떨어지기 까지 약 8초의 시간이 소요된다. 따라서 데이터베이스에서 데이터를 호출할 때, 아두이노 값과, Discriminator의 값은 8초의 시간 차이를 보인다. 또한, 판단의 신뢰성을 향상시키기 위하여, 한 시점의 에러 여부를 판단하는 것이 아닌 4회 연속 에러 라고 판단할 시 불량 의사결정을 내린다. 만약 1~3회 연속 에러 판단을 했을지라도, 4회가 만족되지 않으면 누적 횟수는 다시 0으로 초기화 되고 인식을 계속해서 진행한다.

의사결정의 전반전인 프로세스는 다음과 같다. 첫 번째로, 펀칭 소리의 유무를 판단한다. 소리가 나지 않으면서 스트랩이 낙하 하는 경우, 펀칭시점과 스크랩 낙하의 시간차이로 인한 현상으로서 정상이라고 판단한다. 소리가 나지 않으면서 스크랩도 낙하 하지 않는 경우는 작업이 종료됐음으로 판단한다. 펀칭 소리를 정상이라고 판단하면서 스크랩이 8초 이내로 떨어지는 경우는 정상이라고 판단한다. 그리고 펀칭 소리를 정상이라고 판단하면서 8초 이상 스크랩이 낙하 하지 않는 경우에는 에러 라고 판단한다. 또한 편칭 소리를 비정상이라고 판단하는 경우엔 에러 라고 판단한다. 최종적으로, 3개의 조건을 만족하여 에러 라고 판단할 경우 어플리케이션을 통해 작업자에게 알람을 전송한다.



<Picture 11. 의사결정 로직>



< Picture 12. 의사결정>

위의 그림은 의사결정 방법이다. 적외선 인식은 펀칭 소리가 난 뒤에 컨베이어 벨트를 따라 떨어지는데 까지 약 8초의 시간이 소요된다. 따라서 먼저 펀칭 소리 data를 수집하고 8초 뒤에 scrap data를 수집한다. Buffer구간을 두어 그 기간 동안 지속적으로 에러인지 아닌 지 판단을 하며, 에러라고 판단해도 에러 상태가 4번 축적될 때 알람을 보내도록 설계했다.

Error Prevention System(EP System)

펀칭을 할때 금형에 작용하는 요인은 펀칭 횟수와 대상 철판의 두께이다. 펀칭 횟수는 설계도 데이터를 통해 각 금형의 총 펀칭 횟수를 계산할 수 있다. 그리고 펀칭 횟수는 대상 철판의 두께에 따라 가중치를 반영하여 계산된다. 이렇게 계산된 펀칭 횟수를 누적하고 이를 바탕으로 금형을 추천하며, 금형의 파손시점을 누적하여 이를 바탕으로 금형의 파손 시점을 예측한다.

* 누적 펀칭 횟수 수리모형

Punch force에 대한 수리모형(Turret punch press tooling, amada.fr)은 다음과 같다.

(S = Shear strength(전단강도) , t = Sheet thickness , L = Length of cutting edge)

If shear strength is not known, cutting force can be estimated as :

(UTS = ultimate tensile strength of sheet metal(최대인장강도) , t = thickness ,

L = Length of cutting edge)

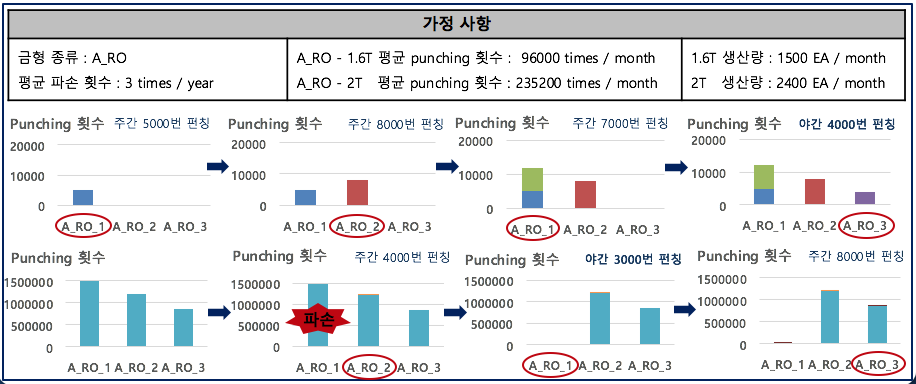
재료가 같은 steel은 UTS 값이 같으므로 punching force 는 두께에 따라 결정된다. 위의 식을 활용하여 계산한 결과, 1.6T 와 2T 를 펀칭하는 힘은 다음과 같다.

**→**

2T를 1번 펀칭하는 것은 1.6T를 1.25번 펀칭하는 것과 같다. 따라서 2T를 생산할 때에는 펀칭 횟수에 1.25의 가중치를 곱하여 계산한 값을 펀칭 횟수로 저장한다.

8.2 금형 추천 프로세스

주간 생산일 경우에는 에러에 즉각 대비할 수 있으므로, 두 번째로 작은 펀칭 횟수의 금형을 추천한다. 야간 생산일 경우에는 에러의 발생 확률을 최소한으로 하기 위해 가장 작은 펀칭 횟수의 금형을 추천하도록 설계한다. 매번 생산량을 입력할 때마다 26개의 금형을 바꾸는 것은 비효율적이므로, 주간과 야간에 한 번씩 총 1일 2회 금형을 추천한다. A\_RO 금형의 예시는 다음과 같다.



< Picture 13. A\_RO 예시>

처음 주간 생산시, A\_RO\_1 금형을 추천한다. 그 다음 주간 생산에는 두 번째로 펀칭 횟수가 작은 A\_RO\_2금형을 추천한다. 그 다음 주간 생산에는 펀칭 횟수가 두 번째로 작은 A\_RO\_1 금형을 추천한다. 야간 생산에는 펀칭 횟수가 가장 작은 A\_RO\_3 금형을 추천한다. 시간이 지나, 펀칭 한도에 가까워져 가장 많이 펀칭한 A\_RO\_1 금형이 파손되면, 금형을 교체하고 다시 A\_RO\_1 금형을 야간에 쓰이는 금형으로 추천한다.

**프로젝트 결과물**

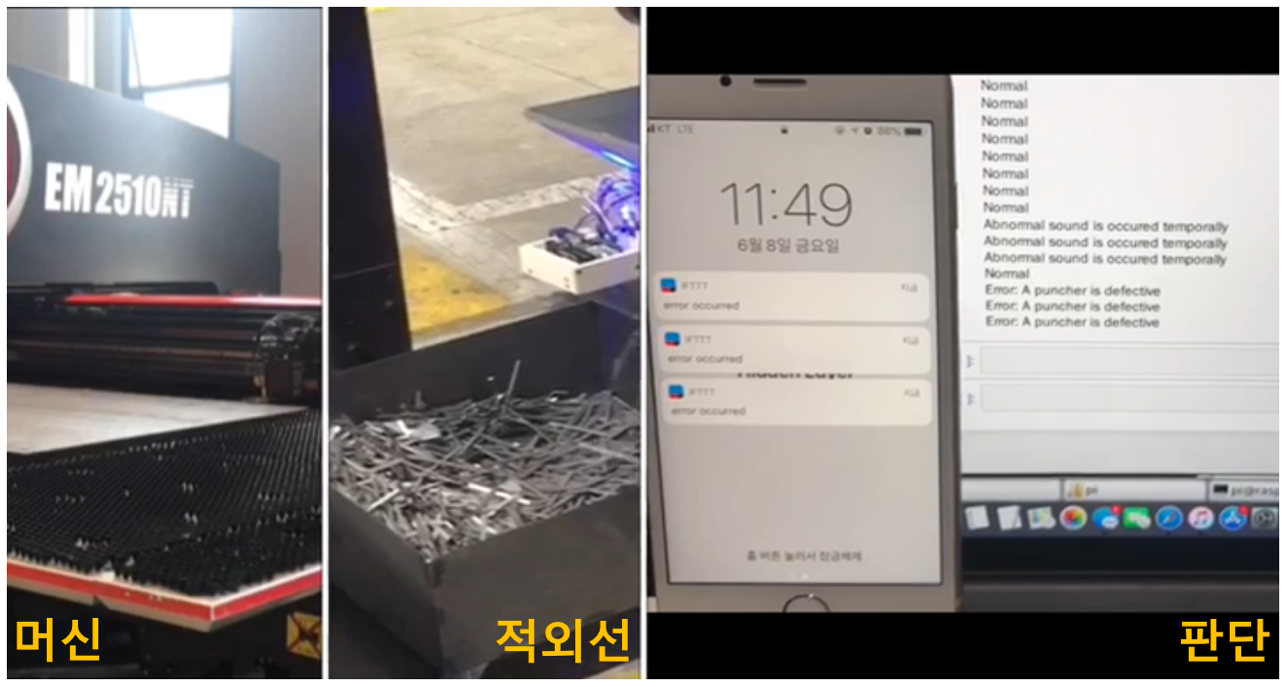
Error Catcher System

다음은 실제 포스-벽진 생산라인에서 EC 시스템을 검증한 모습이다.



<Picture 14. 정상 환경일 때>

정상 환경의 경우, 판단 화면에서 Normal로 표시되며 정상이라고 판단하며 알람을 보내지 않는다.

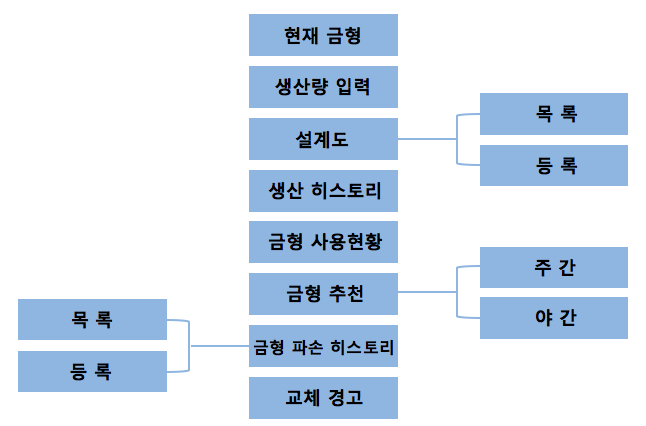


<Picture 15. 비정상 환경일 때>

하지만 비정상 환경일때에는 판단 화면에서 “Abnormal sound is occurred temporarily”로 표시되며 비정상이라 판단한다. 그리고 4회 반복될 시에는 Error 로 판단하여 스마트폰으로 알람을 송신한다.

Error Prevention System

어플리케이션에서는 설계도 자료를 바탕으로 금형 펀칭 횟수를 저장하고, 누적된 펀칭 횟수를 바탕으로 금형을 추천하고 파손을 경고한다. 메뉴 구조도 및 주요 페이지 형태는 다음과 같다.

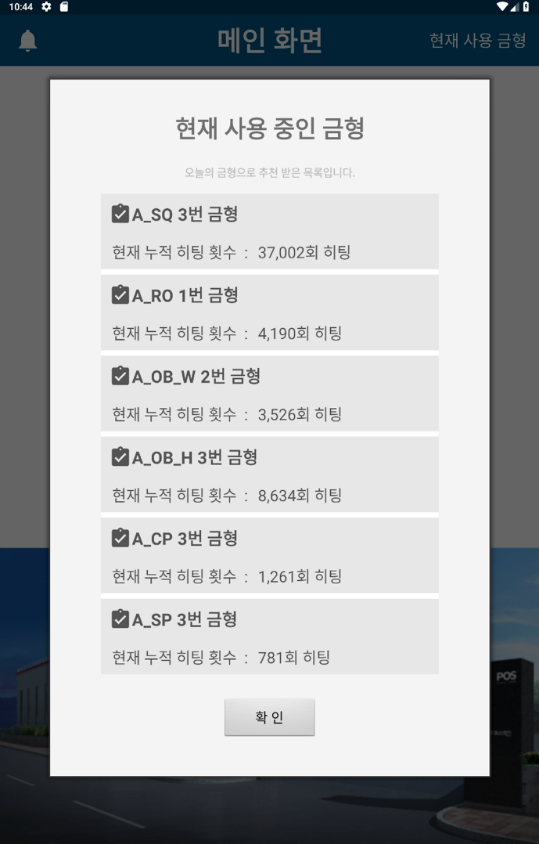
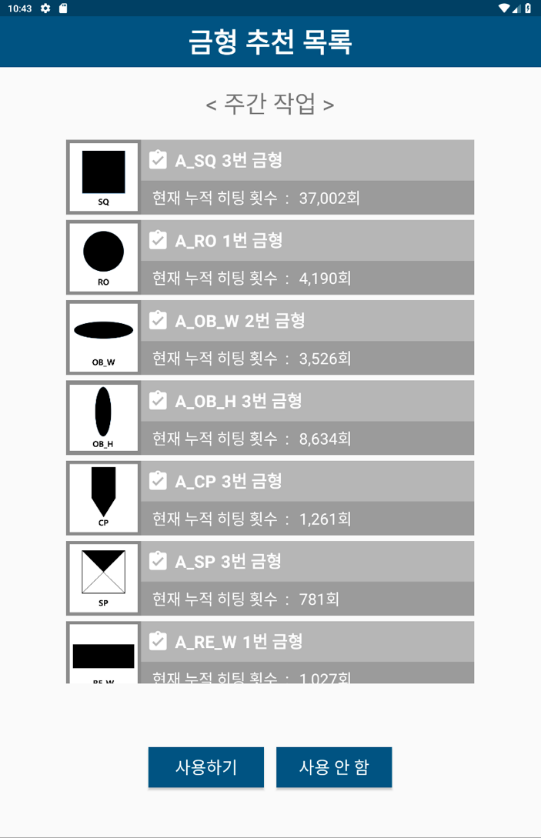


<Picture 16. 메뉴 구조도>



<Picture 17. 앱 메인 화면>

먼저 메인 화면이다. 크게 6가지 메뉴가 있다. 좌측 상단은 금형 교체 경고 알람이며, 우측 상단은 현재 사용하고 있는 금형을 알려준다.



<Picture 18. 금형 추천(주간)> <Picture 19. 현재 사용 중 금형>

금형 추천 화면(주간)은 각 금형 마다 2번째로 펀칭 횟수가 적은 금형들을 추천한다. 이는 야간 화면과 동일하며 야간에는 가장 작은 펀칭 횟수의 금형을 추천한다.



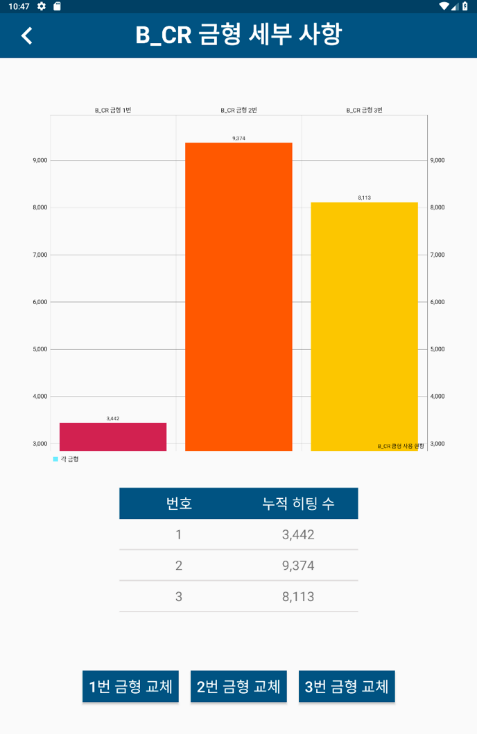
<Picture 20. 생산 정보 입력> <Picture 21. 생산 정보 확인>

생산 정보 입력화면이다. 작업 할 설계도와 생산량, 생산 날짜, 생산 시간, 마지막으로 주간 작업과 야간 작업을 선택한다. 확인 버튼을 누르게 되면, 설계도 별로 필요한 금형이 나오고 판 두께, 총 생산 소요시간이 나오게 된다. 이 정보는 생산 히스토리 메뉴에서 볼 수 있다.



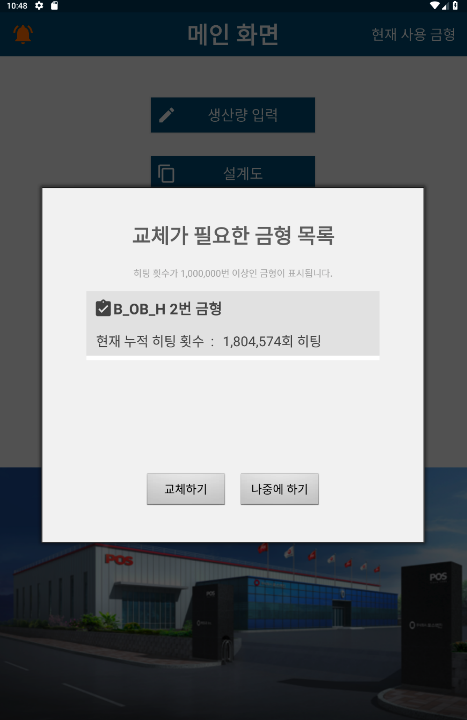
<Picture 22. 설계도 목록> <Picture 23. 설계도 등록>

DataBase에 저장되어 있는 설계도 목록 화면이다. 새로운 설계도를 등록할 수 있도록 설계도 등록 기능도 설계하였다.



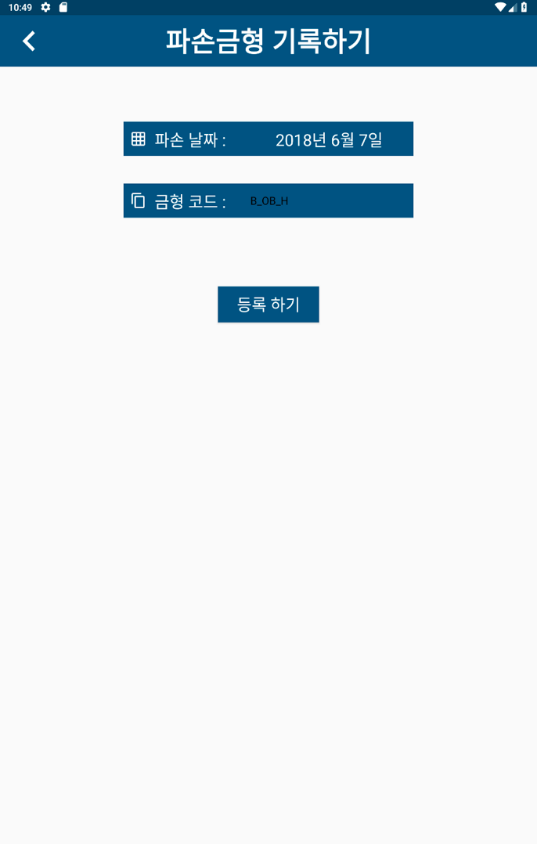
<Picture 24. 금형 사용현황 목록> <Picture 25. 금형 사용현황 세부>

금형 사용현황 목록화면이다. 각 금형의 그림을 누르게 되면 <Picture 43> 과 같이 펀칭의 횟수 여부를 그래프로 나타내 준다. 또한 이 화면에서 펀칭 한도가 가까워지면 바로 교체할 수 있게 교체 기능도 추가하였다.



<Picture 26. 금형 교체 경고>

금형 교체 경고 화면이다. 펀팅 한도에 가까운 금형을 교체하라는 알림을 보낸다.



<Picture 27. 금형 파손 히스토리> <Picture 28. 금형 파손 등록>

금형 파손 히스토리 화면이다. 지금까지 파손된 금형의 종류와 파손됐을 때 펀팅 횟수의 정보를 알려준다. 또한 금형이 파손했을 때, 따로 등록할 수 있도록 금형 파손 등록 기능을 만들었다.

**기대 효과**

프로젝트 기대 효과

포스-벽진 자동화 시스템 개선

- 금형 추천 시스템을 구현하여 금형 파손을 예측

- 추천 알고리즘을 통해 보다 더 효율적으로 금형 사용

- 적외선 인식을 사용하여 작업자에게 scrap이 떨어지는 유무를 시각적으로 보여주어 편리성을 제공

- 에러 알람 어플을 통해 무인 환경에도 생산 가능성을 제공했다.

음향 인식을 통한 Defect 검출

- Machine Learning을 통해 비정상을 감지 할 수 있는 시스템 구축

향후 과제 및 최종 결과물 활용 계획

|  |  |  |
| --- | --- | --- |
| **번 호** | **활용계획** | **비고** |
| **1** | 시스템 도입 | 실제 공장에 시스템 도입 |
| **2** | 논문 작성 | 4차 산업혁명에 따른 공장 개선을 주제로 논문 작성 및 제출 |

<Table 6. 최종 결과물 활용 계획>

**참고문헌**

김성진, 2018, 『코딩셰프의 3분 딥러닝 케라스맛』, 한빛미디어

안드레아스 뮐러, 세라 가이도, 2017, 『파이썬 라이브러리를 활용한 머신러닝』,

한빛아카데미

이근우, “라즈베리파이에 MySQL 설치”, 2017, http://www.ktman.pe.kr/RaspberryPi/5

9483

이시이 모루나, 에사키 노리히데, 2016, 『모두의 라즈베리파이with Python』, 길벗

최근우, “음악신호 머신러닝 초심자를 위한 가이드”, 2016, http://keunwoochoi.blo

gspot.kr

태스, 아두이노(Arduino) 프로그래밍 4-LED제어 해보기1 While 문, 2017, https://blog.

naver.com/mchoi0602/220918629042

Amada-outillage, “Turret punch press tooling”, http://www.amada.fr/media/

user/doc\_655.pdf

Hris Donahue, Julian McAuley , Miller Puckette, 2018, “Synthesizing Audio with Generative Adversarial Networks”

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, Aaron Courville, 2017, “Improved Training of Wasserstein GANs”

Knight of Pi a Raspberry Pi blog, “Installing jupyter on a raspberry pi for notebooks,

debugging and data analysis”, 2017, http://www.knight-ofpi.org/installing- jupyter-on- a-raspberry-pi-for-notebooks-debugging-and-data-analysis/

Kocoa fab, “동전에 반응하는 저금통 만들기”, 2014, https://kocoafab.cc/

make/view/173

Martin Arjovsky, Soumith Chintala, L ́eon Bottou, 2017, “Wasserstein GAN”

Naotokui, “Specgan-generate audio with adversarial training”, 2018, https://github

.com/naotokui/SpecGAN

Pinkwink, “내가 작성한 python 코드에서 내 폰으로 문자 알림을 보내는 방법-ifttt”, 2018, <http://pinkwink.kr/1100>

Raspberrypi.org, “How to install pyaudio”, 2012, https://www.raspberrypi.org

/forums/viewtopic.php?t=25173

Raspberrypi.org, “Installing ipython and jupyter on raspi”, 2015, https://www.

raspberrypi.org/forums/viewtopic.php?t=124936

Scikit learn, “Advenced installation instructions”, 2018, http://scikitlearn.org

/stable/developers/advanced\_installation.html

Simon monk, 2015, 『Raspberry Pi cookbook』, 한빛미디어

**[별 첨]**



- 안전 재고 : 80EA

- 도색 공정 후 생산 LT : 20min

(조립-포장)

- 조립 공정 LT : 15min

- 긴급 주문 횟수 : 10times / month

- 거래처의 납기 기한 :

표준 제품 (4days)

설계 필요 제품 (7days)

비표준 제품 및 외주 진행 제품 (10days)

- 월 평균 판매량 : 700EA

- 일 평균 생산대수 : 30EA

- 일 평균 생산 금액 : 3,000,000 (￦)

(RACK제조비용 평균 : 100,000￦/EA)

<Picture 29. 회사 조감도>

|  |  |
| --- | --- |
| 기업명 | (주) 포스-벽진 |
| 기업 형태 | 중소기업 |
| 소재지 | 경기도 이천시 신둔면 원적로 419-56 |
| 사업 분야 | 통신용 금속 케비넷 RACK /통신장비 제조업체 |
| 종업원 수 | 23명 |
| 연 매출액 | 72억원 (2018.01 기준) |
| 자본금 | 10억원 (2017.09 기준) |
| 주요 거래처 | 통신사 ( LG / KT / SK ) , 삼성 등 |

<Table 5. (주)포스 벽진 기업 정보 표>

<Picture 30. 설비 종류> <Picture 31. 주요 제품 >

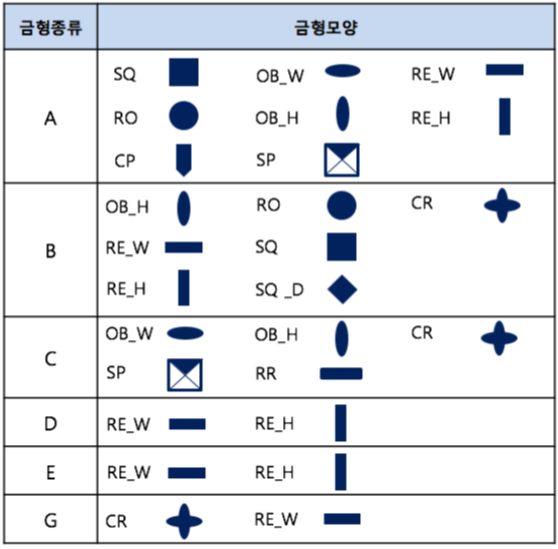
|  |  |  |
| --- | --- | --- |
| **공정 순서** | **사 진** | **내 용** |
| 1 | - | **<고객 주문>**  고객이 영업부와 함께 RACK을 설계하고  주문을 한다. |
| 2 | ../사진/철판.jpeg | **<원자재(철판) 입고>**  포스코 & 현대제철에서 규격에 맞는 철판을 입고한다. |
| 3 | - | **<입고 검사>**  맞는 철판이 입고 되었는지, 불량은 없는지 확인하는 절차를 진행한다. |
| 4 | ../사진/펀칭머신3.jpeg../사진/가공%20후.jpg | **<Punching 공정>**  구멍을 뚫는 Punching 공정을 설계도면에 따라 진행한다. |
| 5 | ../사진/절곡3.jpeg../사진/절곡%20후%20.jpeg | **<절곡 공정>**  Punching 공정이 완료된 철판은 절곡 공정 라인에 들어가게 된다. |
| 6 | ../../../Desktop/스크린샷%202018-03-13%20오후%209.37.11.png | **<용접>**  절곡이 완료된 철판들은 용접공정을 통해  기본 부품으로 완성된다. |
| 7 | - | **<공정 검사>**  모든 공정들이 불량없이 진행되었는지 검사 |
| 8 |  | **<도색 (외주)>**  가공 공정이 완료된 제품은 외주로 보내  도색 과정을 거친다. |
| 9 |  | **<조립>**  도색이 완료된 부품은 설계도면에 맞게  부품을 조립하여 완제품을 만든다. |
| 10 |  | **<포장>**  완제품은 비닐로 포장한다. |
| 11 | - | **<보관 및 출하>**  출하돼야 할 제품들은 박스 포장을 한 뒤, 완제품 재고에 보관한다. |

<Table 6. 공정 순서도>

- 금형 종류



<Picture 32. 금형>



<Table 7. 금형 모양>

평균 근무 일수.: 22 days / 1month

평균 야간 일수 : 15 days / 1month

평균 에러 횟수 : 2.5 times / 1month

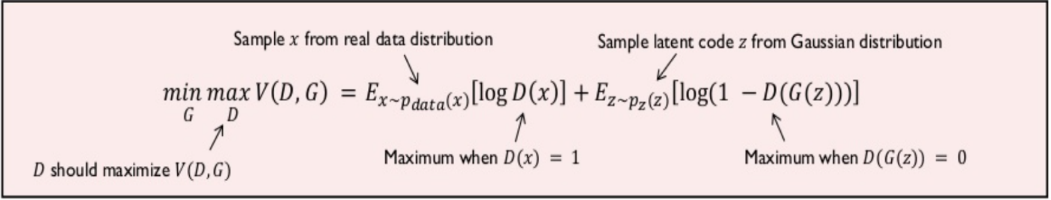
금형 재고 : 2~3개 안전 재고 보유

- 철판 종류

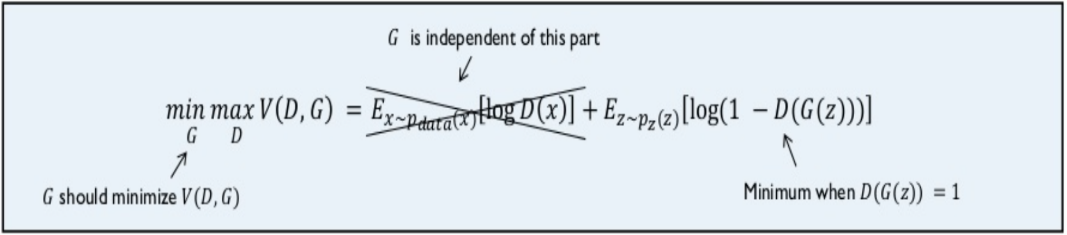
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **1T** | **1.2T** | **1.6T** | **2T** |
| **용도** | 통풍구 | 문 | 바닥 | 기둥 |
| **설비 종류** | 반자동 머신 | | 자동 머신 | |
| **가격(**￦**)** | 23,000 | 26,000 | 31,000 | 36,400 |
| **무게(kg)** | 29.8 | 32.5 | 37.3 | 46.7 |

<Table 8. 철판 종류 표>

- Generate Adversarial Network (GAN) Objective Function



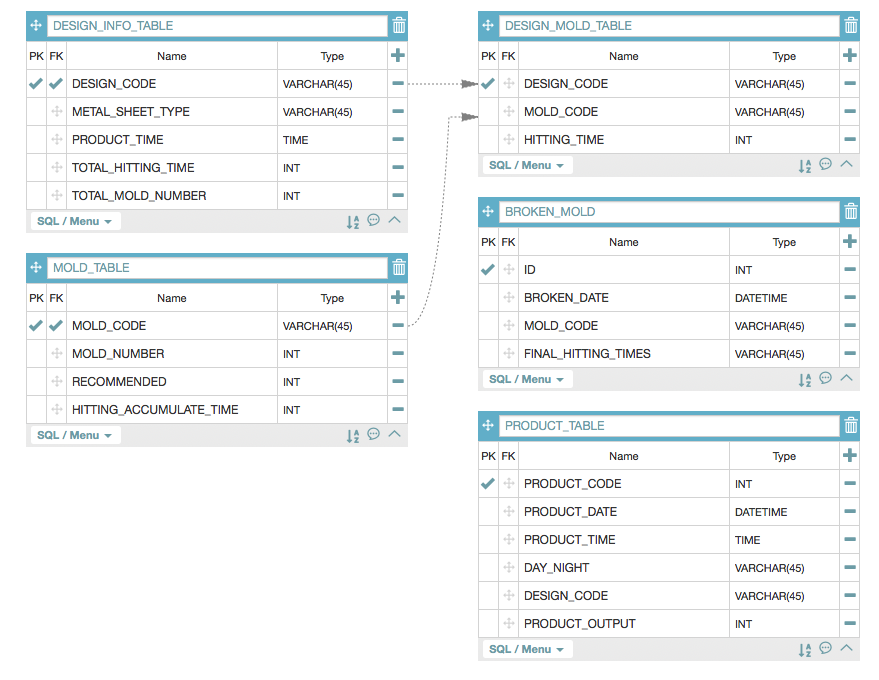
→ Discriminator should maximize



→ Generator should minimize

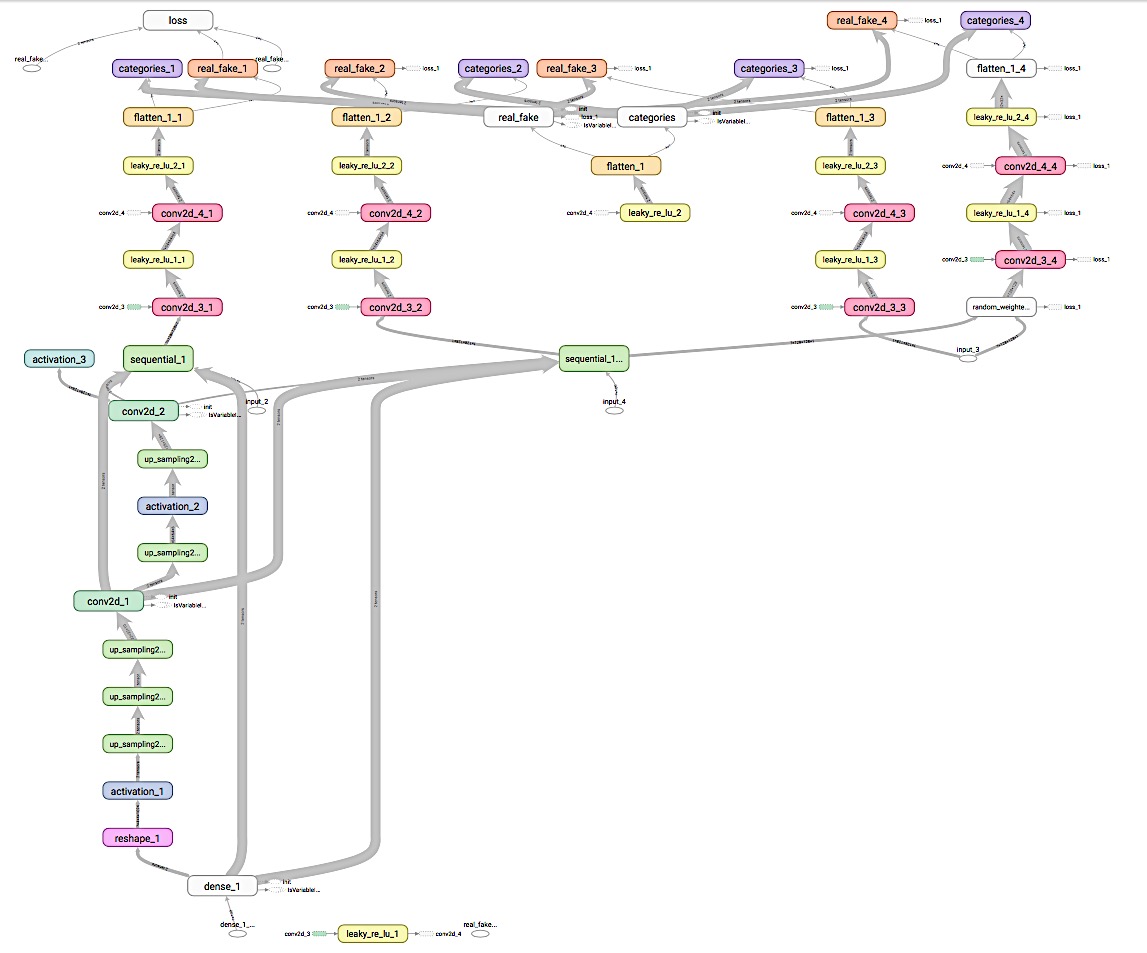
<Picture 33. GAN>

- Error Prevention System ERD



<Picture 34. 시스템 ERD>

- Hidden Layer 구조도



<Picture 35. Hidden layer 구조도>