**최종 보고서**

|  |  |
| --- | --- |
| **과제** | **이미지 노이즈 제거 클래스 분류기** |
| **수업** | **머신 러닝 (4-6)** |
| **지도교수** | **김인겸 교수님** |
| **팀** | **셀프러닝** |
| **연구참여자** | **팀장 김찬영(20190895) : 프로젝트 총괄 및 보고서 작성** |
| **팀원 우상욱(20190919) :** **프로그래밍, 회의록 작성, ppt 제작** |
| **팀원 지성원(20190948) : 프로그래밍, 자료조사** |
| **팀원 허진환(20190954) : 프로그래밍 총괄** |
| **제출일자** | **2023.06.12** |

1. **설계과제 제목 및 소개**

텍스트, 스크린샷, 그래픽 디자인, 그래픽이(가) 표시된 사진

자동 생성된 설명

이미지 노이즈 제거 클래스 분류기

머신 러닝을 통하여 노이즈가 포함된 이미지를 classification하는 모델을 설계한다. 성능의 증가를 위해 오토인코더를 사용하여 이미지의 노이즈 제거 후, '헬리콥터', '비행기', '얼굴'과 같은 101가지 다른 클래스의 이미지를 분류하기 위한 알고리즘을 개발한다. Sequential 모델을 사용하여 인공적으로 특징을 추출하고, 모델을 훈련하여 새로운 입력 이미지의 클래스를 분류하고 예측하는 것을 목표로 한다. 이미지의 특징을 추출하고 학습하기 위해 합성곱 신경망(CNN)을 사용하고, 모델을 분류하기 위해 훈련시킨다. 두 가지 다른 활성화 함수의 영향을 비교할 예정으로, Rectifier (relu) 활성화 함수와 tanh 활성화 함수의 정확도를 비교할 예정이다.

1. **설계과제 추진 목적**

2.1 합성곱 신경망(CNN) 기본 개념 이해

2.2 인공지능에서의 파이썬 활용

2.3 CNN 기반의 오토인코더 모델 설계

2.4 오토인코더 모델을 통한 이미지 잡음 제거

2.5 CNN 기반의 이미지 클래스 분류기 모델 설계

2.6 이미지 클래스 분류기 모델을 통한 이미지 클래스 분류

2.8 오토인코더와 이미지 클래스 분류기 모델의 합성

2.9 CNN 활용 범위와 필요성에 대한 인식 제고

1. **설계과제의 필요성**

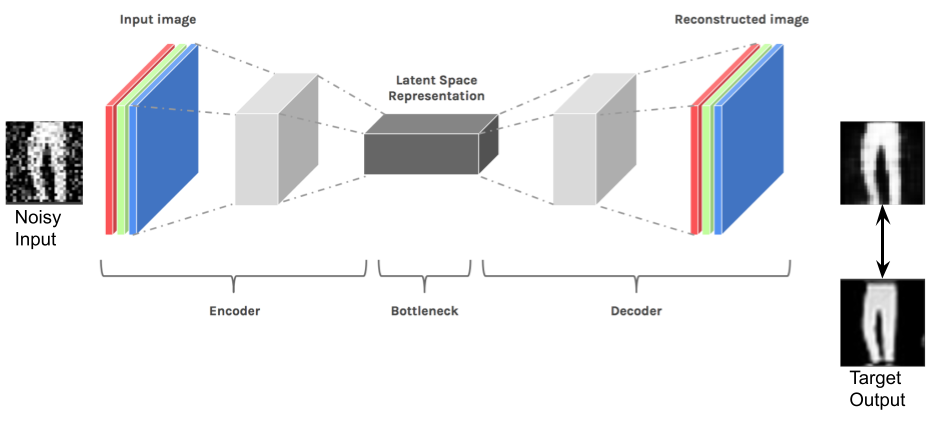
다양한 영상 처리에 있어 이미지 데이터의 처리는 영상처리의 기본으로, 통신 과정 중 데이터에 끼는 노이즈는 항시 존재한다고 할 수 있다. 노이즈가 낀 데이터를 수신해도 해당 데이터를 제대로 해석하여 수신한다면, 통신 과정에서 비용을 들이지 않고도 BER 성능을 높일 수 있으므로, 데이터를 처리하고 분류하는 데 있어 cnn기반의 오토인코더 모델을 통해 노이즈를 제거한 모델을 S/W적으로 구현하는 것은 H/W적으로의 현저한 비용 절감을 가져올 수 있다. 또한, 이렇게 수신한 데이터를 클래스화하여 분류하는 작업을 통해 데이터 분류에 소모되는 인력 및 시간을 절약할 수 있고, 더욱 다양한 방향으로 데이터가 이용되어질 수 있으며, 다양한 문제 해결 (의료 진단 또는 자율주행차에서의 표지판 인지) 등에 쓰일 수 있으므로 설계 과제를 추진하게 되었다.

1. **설계과제의 목표**
   1. 이미지 데이터셋 (cifar-10) 로드 및 잡음 추가
   2. 잡음 제거 기능 오토인코더 모델 설계
   3. 오토인코더 모델 훈련을 통한 예측 (잡음 제거 수행)
   4. 이미지 클래스 분류기 모델 설계
   5. 이미지 클래스 분류기 모델 훈련을 통한 예측 (이미지 클래스 분류 수행)
   6. 오토인코더 모델과 이미지 클래스 분류기 모델의 합성

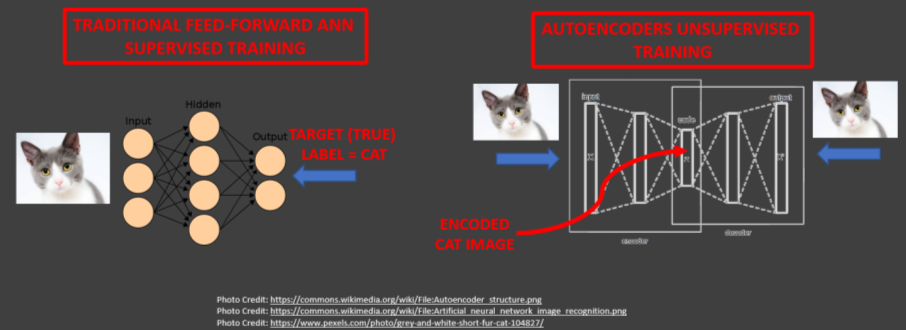
|  |  |
| --- | --- |
| **현실적 제한 요소** | **내용** |
| **경제** | 모델 훈련에 사용되는 H/W 비용에 대한 문제를 인식하여야 한다. |
| **편리** | 이미지 처리에 있어 기존 사람이 하던 방식만큼 정확하고, 사람이 하던 방식보다 편리해야 한다. |
| **윤리** | 활용되는 CNN 이미지 처리 기술이 불법적이나 비윤리적으로 이용되어서는 안된다. |
| **사회** | 인공지능의 발달로 인한 일자리 감소 문제를 인식하여야 한다. |

1. **설계과정**
   1. **설계 기초이론**

5.1.1오토인코더: 오토인코더는 데이터 인코딩(representation learning) 작업을 수행하는 인공 신경망의 한 유형으로, 오토인코더의 특징으로는 입력 데이터를 입력 및 출력으로 사용한다.



5.1.2 코드 레이어: 오토인코더는 네트워크에 병목 현상을 만들어 줌으로서 동작한다. 이 병목 현상은 네트워크가 원래 입력의 압축된(encoded) 버전을 생성하도록 한다. 오토인코더는 입력 데이터와 상관관계가 있는 경우에 잘 작동한다.(입력 데이터가 모두 독립적인 경우에는 잘 작동하지 않을 수 있음)

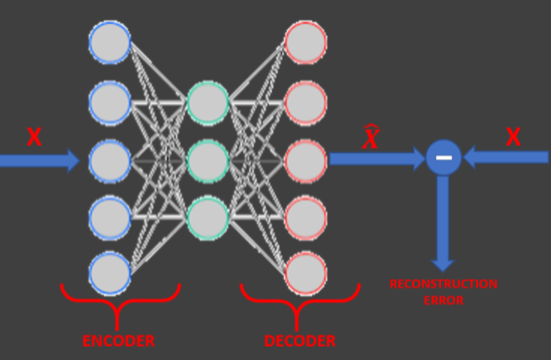


5.1.3 오토인코더의 수학적 원리:

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

5.1.4 재구성 오차: 오토인코더의 목표는 입력 X와 네트워크 출력 X^ 사이의 재구성 오차를 최소화하는 것이다. 오토인코더의 차원 축소(잠재 공간)는 선형 활성화 함수를 사용하는 경우 PCA(주성분 분석)와 매우 유사하다.



5.1.5 신경망: 신경망의 기본 단위는 퍼셉트론이다. 입력 레이어에서 이 퍼셉트론으로 입력이 들어가고, 이 퍼셉트론은 결과를 출력한다. 본 프로젝트에서의 입력은 이미지로 들어가게 되는데, 이 이미지는 훈련 및 테스트 데이터셋으로 나눠진다. 그런 다음 훈련 데이터셋으로 모델을 학습시킨다. 그 후 테스트 데이터셋으로 모델의 성능을 테스트한다. 프로세스의 마지막 부분은 임의의 입력 데이터의 클래스를 예측하는 것이다.

5.1.6 입력층에서의 이미지 정규화: 첫 번째 단계는 입력 이미지를 정규화하는 것으로, 이미지는 RGB 값의 집합으로 이루어져 있으며, 이 값들은 채널로써 주어진다. 채널의 값이 1이면 그레이스케일, 3이면 RGB이다. 따라서 이러한 값을 공통 범위로 정규화해야 한다. 첫 번째 레이어에서 이미지를 RGB로 정규화한다.

5.1.7 활성화 함수: 입력을 임계값으로 처리하기 위해 ReLU를 사용한다. 이는 일반적으로 합성곱 계층에서 정의된다. 텐서플로우에서는 두 가지 방법으로 정의할 수 있는데, 첫 번째는 합성곱 계층에서 정의하는 것이고, 두 번째는 활성화 함수를 포함하는 다른 계층을 추가하는 것이다. 본 프로젝트에서는 첫 번째 방법인 합성곱 계층에서 정의하는 옵션을 사용하기로 하였다. ReLU의 공식은 다음과 같다 : f(x)=max⁡(0,x)

5.1.8 합성곱 계층: 합성곱 계층은 입력 위에 일련의 필터를 합성한다. 높은 필터 응답은 필터와 입력 간의 유사성을 나타내고, 그 반대도 마찬가지이다. 이 계층에서 얻은 필터 출력을 통해 입력 이미지의 클래스에 대한 결정을 내릴 수 있다! 이 계층은 입력 이미지의 차원은 변경하지 않으며 출력 이미지의 채널 수를 변경하는 선형 변환을 수행한다. 합성곱 계층은 일정한 바이어스로 입력을 받고 각 입력에 대하여 가중치가 정의된다. 이 가중치는 에러 값을 역전파하여 모델을 개선하기 위해 조정할 때 오류를 줄이는 데 도움이 된다.

5.1.9 Pooling층, Maxpool 계층: Maxpool 작업은 이미지를 다운샘플링하기 위해 사용된다. 이는 입력을 가져와 원하는 크기로 변환한다. 이 단계에서는 행과 열의 수를 변경하지만 깊이는 동일하게 유지된다. Maxpool작업은 데이터에 과적합되지 않도록 도움이 된다.

5.1.10 완전 연결 계층: 이 계층은 모든 합성곱, Maxpool 및 ReLU 작업이 완료된 후 사용된다. 이 계층은 위의 작업이 출력인 주어진 입력의 숫자 값을 계산한다.

5.1.11 softmax 계층: 합성곱 신경망의 마지막은 일반적으로 softmax 분류기 또는 SVM(Support Vector Machine)이다. 본 프로젝트에서는 softmax를 사용할 예정이다. softmax 분류기는 배열을 입력으로 받고, 데이터셋의 다른 범주에 대한 출력을 제공한다.

**5.2.1 소프트웨어 기능블록도**

**5.2.2 기능적 요구사항 명세서**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **요구사항** | **내용** | **설명** | **우선순위** |
| **No\_01** | **이미지 데이터셋 로드** | **이미지 노이즈 추가/제거 및 클래스화를 위한 이미지 데이터셋 로드** | 클래스가 10개인 이미지 데이터셋인 Cifar-10 데이터셋을 훈련용/검증용 데이터로 나누어 로드한다. | **1** |
| **No\_02** | **이미지 잡음 추가** | 로드된 cifar 데이터셋에 이미지 잡음 추가 | 오토인코더의 입력 데이터로 들어가 잡음 제거가 수행되도록 하기 위해 임의의 노이즈 강도를 설정해 각 이미지에 잡음을 추가한다. | **2** |
| **No\_03** | **오토인코더 모델 설계** | 이미지 잡음 제거를 위한 오토인코더 모델 설계 | ReLU 활성화 함수를 사용한 합성곱층과 이미지 크기를 1/2로 줄이는 풀링층 이후 다시 합성곱층으로 이어지는 인코더 부분과 이미지 크기를 원복하고, 합성곱층을 사용하여 이미지 특성 추출하여 압축으로부터 복원하는 디코더 부분을 가진 오토인코더 모델을 설계한다. | **3** |
| **No\_04** | **오토인코더 모델 훈련** | 설계된 오토인코더 모델 훈련 | 잡음이 추가된 이미지 훈련용 데이터에 대해 원본 훈련용 데이터를 비교를 위한 정답 데이터로 사용해 오토인코더 모델을 훈련한다. | **4** |
| **No\_05** | **잡음 제거 이미지 출력** | 잡음이 제거된 이미지 출력 | 잡음이 추가된 이미지가 오토인코더 모델 훈련을 통해 예측 결과로 잡음 제거 이미지를 반납하고, 이를 정답 데이터(원본 훈련용 데이터)와 같이 출력함으로써 잡음 제거 수행 결과를 확인한다. | **5** |
| **No\_06** | **잡음 제거 이미지 데이터화** | 잡음이 제거된 이미지 훈련용/검증용 분류 | 잡음이 추가된 이미지에 대해 이미지 분류를 수행해야 하므로 훈련용/검증용 데이터로 나누어야 한다. | **6** |
| **No\_07** | **이미지 클래스 분류기 모델 생성** | 잡음 제거 이미지 데이터셋에 대한 이미지 클래스 분류를 위한 이미지 분류기 모델 생성 | ReLU 활성화 함수를 사용한 합성곱층과 풀링층을 3번 통과하고 차원 축소를 거친 후 완전 연결층으로 이어진 후 확률 값을 출력하는 이미지 클래스 분류기 모델을 생성한다. | **7** |
| **No\_08** | **이미지 클래스 분류기 모델 훈련** | 설계된 이미지 클래스 분류기 모델 훈련 | (잡음 제거 과정을 거친)훈련용 데이터에 대해 정답 데이터인 | **8** |
| **No\_09** | **클래스 분류 결과 출력** | 잡음 제거 과정을 거친 데이터의 클래스 분류 결과 출력 | 잡음 제거 과정을 거친 데이터의 클래스 분류 결과를 이미지와 예측/정답 클래스가 함께 나오도록 출력 | **9** |

1. **제작(Implementation)**
   1. **제작과정**

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

이미지 분류를 위해 다양한 카테고리를 가진 이미지 dataset인 cifar10을 사용하고, 훈련용/검증용 데이터로 구분하였다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

스크린샷, 다채로움, 예술, 모자이크이(가) 표시된 사진

자동 생성된 설명

이미지 데이터셋이 제대로 로드되어 훈련용 데이터로 들어갔는지 확인하기 위해 데이터 이미지를 랜덤한 인덱스로 접근하여 위와 같이 15\*15 형식으로 출력(출력 화면 밑으로 스크롤 시 15 \* 15 형식)하였다.

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

이미지 노이즈 추가를 위해 노이즈 강도 설정 후 각 이미지 데이터 픽셀에 노이즈를 추가하고, 노이즈가 잘 추가되었는지 확인하기 위해 노이즈 추가된 이미지 한 장을 임의 출력하였다.

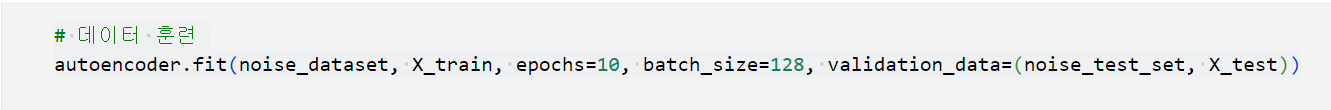
텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

오토 인코더 모델의 인코더 부분을 설계할 때 이미지 데이터 특성을 추출하기 위해 Conv2D 함수를 이용하고, MaxPooling2D()함수를 사용하여 차원 축소 및 압축을 시행하도록 하였다. 디코더 부분을 설계할 때는 그의 역순으로 Upsampling2D() 함수를 이용하여 차원 복원 및 압축으로부터의 복원을 시행하도록 하였다.



설계한 오토 인코더 모델을 통해 데이터 훈련을 fit() 함수를 통해 진행하였다. 학습 반복 횟수인 epochs를 10으로 두었고, 노이즈 추가 데이터 noise\_dataset 입력에 대해 정답 레이블인 원본 데이터 X\_train과 비교하여 학습하도록 하였다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

모델 성능 측정을 위해 정확도 파라미터를 사용하여 모델 성능을 측정하였고, 해당 모델에서 0.566의 정확도를 보였다. 또한, predict()함수를 통해 입력데이터 noise\_test\_set[:10] (10개)에 대한 오토인코더 모델 출력의 예측 결과를 predicted에 담고 아래와 같이 예측 결과를 출력하였다.

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

noise\_test\_set[:10] (10개)에 대한 오토인코더 모델 출력의 예측 결과인 predicted를 위와 같이 출력하였다. 1행에는 노이즈 추가된 이미지인 noise\_test\_set[:10] (10개)가 출력되고, 2행에는 그에 대한 오토인코더 모델 출력의 예측 결과인 predicted 이미지 10개가 출력된다.

콜라주, 텍스트이(가) 표시된 사진

자동 생성된 설명

**6.2 제작 시 문제점 및 개선사항**

1. **시험**

**8. 평가**

**8.1 정량적/정성적 목표달성도 평가**

|  |  |  |  |
| --- | --- | --- | --- |
| **항목** | **목표** | **달성률** | **비고** |
| **1** |  |  |  |
| **2** |  |  |  |
| **3** |  |  |  |
| **4** |  |  |  |
| **5** |  |  |  |
| **6** |  |  |  |

**8.2 현실적 제한요소 달성도 평가**

|  |  |  |
| --- | --- | --- |
| **현실적 제한 요소** | **내용** | **달성 결과** |
| **경제** | 모델 훈련에 사용되는 H/W 비용에 대한 문제를 인식하여야 한다. |  |
| **편리** | 이미지 처리에 있어 기존 사람이 하던 방식만큼 정확하고, 사람이 하던 방식보다 편리해야 한다. |  |
| **윤리** | 활용되는 CNN 이미지 처리 기술이 불법적이나 비윤리적으로 이용되어서는 안된다. |  |
| **사회** | 인공지능의 발달로 인한 일자리 감소 문제를 인식하여야 한다. |  |

**8.3 기능적 요구사항 달성도 평가**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **요구사항** | **내용** | **설명** | **달성률** |
| **No\_01** | **이미지 데이터셋 로드** | **이미지 노이즈 추가/제거 및 클래스화를 위한 이미지 데이터셋 로드** | 클래스가 10개인 이미지 데이터셋인 Cifar-10 데이터셋을 훈련용/검증용 데이터로 나누어 로드한다. | **100%** |
| **No\_02** | **이미지 잡음 추가** | 로드된 cifar 데이터셋에 이미지 잡음 추가 | 오토인코더의 입력 데이터로 들어가 잡음 제거가 수행되도록 하기 위해 임의의 노이즈 강도를 설정해 각 이미지에 잡음을 추가한다. | **100%** |
| **No\_03** | **오토인코더 모델 설계** | 이미지 잡음 제거를 위한 오토인코더 모델 설계 | ReLU 활성화 함수를 사용한 합성곱층과 이미지 크기를 1/2로 줄이는 풀링층 이후 다시 합성곱층으로 이어지는 인코더 부분과 이미지 크기를 원복하고, 합성곱층을 사용하여 이미지 특성 추출하여 압축으로부터 복원하는 디코더 부분을 가진 오토인코더 모델을 설계한다. | **100%** |
| **No\_04** | **오토인코더 모델 훈련** | 설계된 오토인코더 모델 훈련 | 잡음이 추가된 이미지 훈련용 데이터에 대해 원본 훈련용 데이터를 비교를 위한 정답 데이터로 사용해 오토인코더 모델을 훈련한다. | **100%** |
| **No\_05** | **잡음 제거 이미지 출력** | 잡음이 제거된 이미지 출력 | 잡음이 추가된 이미지가 오토인코더 모델 훈련을 통해 예측 결과로 잡음 제거 이미지를 반납하고, 이를 정답 데이터(원본 훈련용 데이터)와 같이 출력함으로써 잡음 제거 수행 결과를 확인한다. | **100%** |
| **No\_06** | **잡음 제거 이미지 데이터화** | 잡음이 제거된 이미지 훈련용/검증용 분류 | 잡음이 추가된 이미지에 대해 이미지 분류를 수행해야 하므로 훈련용/검증용 데이터로 나누어야 한다. | **100%** |
| **No\_07** | **이미지 클래스 분류기 모델 생성** | 잡음 제거 이미지 데이터셋에 대한 이미지 클래스 분류를 위한 이미지 분류기 모델 생성 | ReLU 활성화 함수를 사용한 합성곱층과 풀링층을 3번 통과하고 차원 축소를 거친 후 완전 연결층으로 이어진 후 확률 값을 출력하는 이미지 클래스 분류기 모델을 생성한다. | **100%** |
| **No\_08** | **이미지 클래스 분류기 모델 훈련** | 설계된 이미지 클래스 분류기 모델 훈련 | (잡음 제거 과정을 거친)훈련용 데이터에 대해 정답 데이터인 | **100%** |
| **No\_09** | **클래스 분류 결과 출력** | 잡음 제거 과정을 거친 데이터의 클래스 분류 결과 출력 | 잡음 제거 과정을 거친 데이터의 클래스 분류 결과를 이미지와 예측/정답 클래스가 함께 나오도록 출력 | **100%** |

**9. 추진체계**

본 과제의 추진을 위한 팀은 3인으로 구성되며 다음과 같은 역할을 담당한다. 이론 연구 단계에서부터 과제 종료시까지 협력과 보완을 통해 팀의 목표를 달성할 수 있도록 한다.

|  |  |  |
| --- | --- | --- |
| **팀장** | **김찬영** | 프로젝트 총괄 및 보고서 작성 |
| **팀원** | **허진환** | 프로그래밍 총괄 |
| **팀원** | **지성원** | 프로그래밍, 자료조사 |
| **팀원** | **우상욱** | 프로그래밍, 회의록 작성, ppt 제작 |

**10. 설계 추진 일정 :** *2023년 4월 13일 ~ 2023년 6월 18일*

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **수행 내용** | | **일정** | | | | | | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
| **목표와 기준 설정** | -설계목표 설정  **-목표 달성 방법에 대한 논의** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **합성** | **-기능별 구현 방법 결정**  **-적용할 기술** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **분석** | **-세부적인 기능 구현 방법 결정**  **-목표달성 가능성 확인** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **제작** | **-오토인코더 및 분류기 프로그래밍** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **시험/평가** | **-시험 및 검증**  **-재설계** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
| **결과** | **-결과보고 및 시연** |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

**11. 결론**