

2025-2 컴퓨터비전 기말고사 (20241519 조예성)

GUI 프로그램을 활용하여 상호작용하는 것:

환경으로부터 정보를 받아 인식하고 정보 바탕으로 행동하여 환경과 상호작용:

파이썬 위젯 및 GUI 제작 라이브러리

- 위젯 위치 설정: self. (, , ,)

과적합 방지 기법 총칭: 규제

Tensorflow 프로그래밍 핵심 4가지:

한 갈래 모델: | 여러 갈래 모델:

예측과 정답의 오차를 구하는 함수:

- 두 종류: ,

- 로그 활용:

불균형 데이터 해결 위하여 사용하는 것:

- 객체 탐지시 주로 ()에서 문제 발생

가중치 변경에 활용:

주요 방법: , , . ,

적응적 학습률 활용:

- 학습률을 적용하면 함수 내부에서 자동으로 학습률 변경을 하지 않는다: O, X

학습 데이터에서만 잘 작동하는 것: | 학습 데이터조차 잘 작동 못 하는 것:

규제 대표적 3 종류: , ,

데이터 증강 2 종류: ,

이미 잘 작동하는 모델의 가중치를 가져와서 학습:

CNN, Vision Transformer과 같은 특징 추출기 역할의 기본 신경망 구조:

유명한 사전 학습 모델: , ,

- 인셉션 사용하는 모델:

- 인셉션이란?

- 솟컷 이용:

- 효과: 기존 CNN보다

이미지에서 클래스 하나 추출:

이미지에서 위치와 레이블 추출:

물체가 점유하는 픽셀 판별:

영상에서 특정 물체가 사전에 따라 어떻게 이동되는지:

- 한 개만 추적:

- 여러 개 추적:

객체 검출 문제의 성능 측정:

- 설명: 얼마나

- 일반적 임계값 2개: `

정밀도 계산 식: $\text{정밀도} = \frac{\text{TP}}{\text{TP} + \text{FN}}$

재현율 계산 식: $\text{재현율} = \frac{\text{TP}}{\text{TP} + \text{FP}}$

정밀도와 재현율 좌표를 이으면:

- 아래 면적: $A = \frac{1}{2} \times \text{TP} \times \text{FN}$

- 평균 내면:

모션 분석에 필요한 것:

연속된 영상에서 화소의 발길 변화를 기반으로 물체 움직임 추적:

old_gray: I_{old} new_gray: I_{new}

성공 추적 여부:

코너 특징점 검출:

추적 근본 알고리즘:

- 단점:

추적 방식 2개: I_{old} , I_{new}

- 미래 프레임 활용:

- 과거 프레임만 활용:

카메라 하나만 활용 영어:

- 두 종류: I_{old} , I_{new}

- 추적 여러 개:

- 추적 1개:

수초 내지 n분후 다른 카메라에 나타나는 동일 물체 연결/장기 재식별:

Bounding Box 기반 추적:

영역 자체(픽셀 단위) 추적:

추적 처리 과정: $I_{old} \rightarrow I_{new} \rightarrow \text{Tracking Result}$

- 쌍 맺기에서 사용하는 알고리즘:

- 해당 알고리즘이 찾는 것:

매칭 실패시:

- 필요시:

구글 개발 AI파이프라인

사람의 관절 위치를 랜드마크 설정 후 검출:

사람 인식 방법 2개:

- 사람 먼저 찾고 바운딩 박스:
- 이미지에서 일단 관절을 추출:

둘 중 속도 빠른 것:

어텐션 사용:

- 발표 논문:

입력 시퀀스 내에서 관계 동적 학습:

요약 정보 만들기:

요약 정보 활용해 생성:

- 주로 비율: 대

3차원 공간 안의 점들을 모아놓은 데이터 구조:

- 수집 방법:

3D Point Cloud를 입력받아 처리하는 딥러닝 모델:

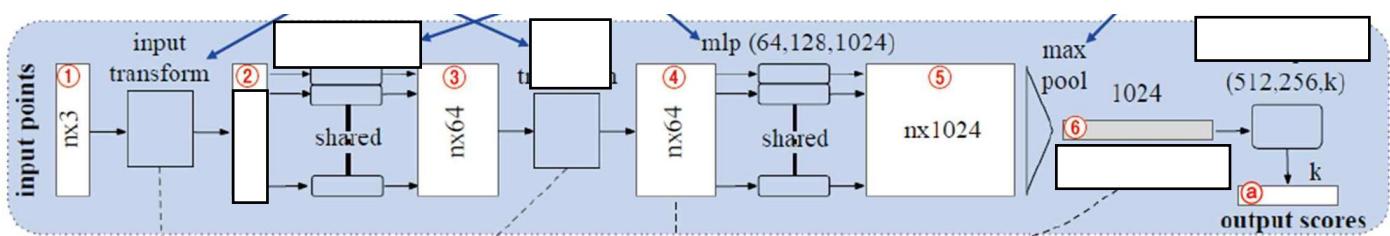
- 용도:

point cloud는 순서가 있다: O, X

pointNet 특징:

사용하는 좌표 구조: $n(\quad, \quad, \quad)$

pointNet 내부 작은 네트워크 구조:



새로운 이미지를 만들어 내는 모델:

대표적 생성 모델:

데이터가 가우시안 분포를 따른다면 () 과 ()만 있으면 새로운 샘플을 쉽게 생성 가능

여러 개의 가우시안을 조합하여 복잡한 분포를 근사하는 모델:

입력 특징을 압축했다 복원:

- 특징 벡터:

- 있는 공간:

맘대로 latent space에서 선택해서 디코더 입력시 이상한 이미지 출력 수정 해결:

- 핵심 2개:

하나의 고정된 latent vector가 아닌()를 출력하게 만듬

생성자와 판별자가 경쟁하며 정교한 이미지 만들도록 학습되는 신경망:

기존 텐서플로우로 감지하는 알고리즘:

코드 암기

```
class BeepSound(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setWindowTitle('삑 소리 내기') # 윈도우 이름과
        self.setGeometry(200,200,500,100)

        shortBeepButton=QPushButton('짧게 삑',self) # 버튼 생성
        longBeepButton=QPushButton('길게 삑',self)
        quitButton=QPushButton('나가기',self)
        self.label=QLabel('환영합니다!',self)

        shortBeepButton.setGeometry(10,10,100,30) # 버튼
        longBeepButton.setGeometry(110,10,100,30)
        quitButton.setGeometry(210,10,100,30)
        self.label.setGeometry(10,40,500,70)

        shortBeepButton.clicked.connect(self.shortBeepFunction)
        longBeepButton.clicked.connect(self.longBeepFunction)
        quitButton.clicked.connect(self.quitFunction)

model=tf.keras.models.load_model('cnn_v2.h5')
```