대본

* 안녕하세요 지능형 소프트웨어 기말 발표를 시작하겠습니다.
* 목차는 다음과 같습니다. 주제 소개 및 선정 이유를 말씀드리고 제가 선정한 주제를 실생활에 어떻게 사용할 수 있는지 활용 방안, 사용한 모델에 대한 설명 그리고 코드 설명 및 수행 순으로 진행하겠습니다.
* 제가 선정한 주제는 이미지 제너레이터입니다. 그림과 글의 상관 관계를 파악하고 그 의미를 이해하는 것은 인간만이 지닌 고차원적인 인지 능력이라고 여겨졌었지만 더 이상 그렇지 않다고 할 수 있습니다. 그리고 저는 이에 대한 방증으로 이미지 제너레이터를 선정하였습니다. 이미지 제너레이터를 설명드리자면 인공지능을 이용해 그림을 자동으로 생성하는 기술입니다. 사람이 텍스트로 명령하면 AI가 학습에 따라 결과물을 내놓는 ‘text to image’ 방식을 사용하고 있습니다. 그림 인공지능의 역사에 관해 간단히 설명드리면 그림 인공지능은 2012년 이미지넷 대회에서 심화신경망 방식의 인공지능을 선보인 이후 비약적인 발전을 거두며 제너레이티브 애드버시럴 넷워크가 2014년에 본격적으로 등장했지만 당시엔 여러가지 애로사항이 있었고 성능의 한계가 극명했기 때문에 대중들의 많은 주목을 받지는 못했습니다. 이후에도 구글의 텐서플로우를 활용한 사진 설명 기능 등, 다양한 인공지능이 등장했지만 접근성의 조금의 발전만 있었을 뿐 단순한 작업만 가능한 정도에 그쳤기 때문에 마찬가지로 대중의 관심을 끌기엔 미비했습니다.
* 하지만 2022년 8월 22일에 출시한 text-to-image 인공지능 모델 스테이블 디퓨전의 등장과 동시에 그림 인공지능의 시대가 열려 이 모델을 채택한 인공지능이 대거 생겨나고 있습니다. 저는 그림 AI의 사용을 커뮤니티 등을 기점으로 유행한 것이 딥러닝, 인공지능 등의 얼핏 어렵게 느껴질 수도 있는 주제를 대중들에게 친숙하게 다가간 좋은 예였다고 생각하고 이미지 제너레이터가 전에 없던 새로운 창조 도구가 될 수 있다고 생각합니다. 동시에 그 사용 방법이 너무나도 쉽고 간단하다는 장점은 매우 많은 가능성을 내포하고 있다고 생각했기 때문에 주제로 선정하게 되었습니다.
* 다음은 활용 방안입니다. 우선 앞서 말씀드린 것처럼 제가 생각하는 이미지 제너레이터의 가장 큰 장점은 사용 방법이 너무나도 쉽다는 점이었습니다. 스테이블 디퓨전 모델의 등장 이후 이를 채택한 많은 인공지능이 등장했고 무료로 시연할 수 있게 제공하는 사이트 또한 많았기에 직접 사용해보았습니다. 각각 세 개의 프로그램에 상명 유니버시티를 입력했을 때 나온 결과물입니다.
* 또한 이미지 제너레이터는 높은 상업적 이용가치를 갖고 있습니다. 해당 기사의 표현을 인용하면 장면 구성이나 광원 효과 등 이미 현업 작가 수준의 작품을 만든다고 평가받는다고 합니다.
* 실제 사용한 예시입니다. 고전 게임에 사용된 퀄리티 낮은 모델링을 스테이블 디퓨전을 통해 개선한 예시입니다. 게임을 포함한 매체에 디자인에 관련된 부분에 더 이상 이전과 같은 많은 시간과 돈을 투자할 필요가 없어질 수도 있을 것으로 보입니다.
* 다음으로 제너레이터를 이용한 그림이 인간만의 고유한 영역이라고 느꼈던 예술에도 관여할 수 있을 것으로 보입니다. 아래 사진은 올해 9월 콜로라도 박람회에서 열린 미술대회에 실제로 출품해 1등을 수여한 작품입니다. 실제로 이미지 제너레이터가 완전한 무에서 유를 창조하는 것이 아니기 때문에 참가 자체에 관한 많은 갑론을박이 있지만 아래 작품의 퀄리티까지는 아무도 부정하지 못하고 있습니다.
* 다음은 사용한 디퓨전 모델에 대해 설명드리겠습니다. 정확히 말해서 그림 AI 모델은 완전한 무에서 유를 창조한다고는 할 수 없습니다. 우선 이름의 단어 디퓨전은 액체나 기체에 다른 물질이 섞이면 번져나가다가 일률적인 농도로 바뀌는 현상을 뜻합니다. 대표적인 예시로 물에 잉크 한방울을 떨어트리는 것을 디퓨전이라고 합니다. 스테이블 디퓨전 모델은 이와 같은 현상에 착안해 이미지에 노이즈를 단계적으로 퍼트리며 입히고 역으로 노이즈를 제거하며 원본 이미지를 복원해 마치 잉크가 DRIP 되기 전의 상태를 만드는 것을 목표로 학습합니다.
* 학습 원리입니다. 왼쪽은 노이즈를 넣는 과정을 표현한 함수와 예시입니다. 어떤 이미지에 노이즈를 만드는 함수 q를 정의했을 때 x0 이미지를 넣으면 약간의 노이즈가 들어간 x1 이미지를 반환합니다. 이걸 반복하면 원본 이미지는 더 이상 이미지의 형태를 확인할 수 없는 노이즈만 남게 됩니다. 오른쪽은 노이즈에서 원본 이미지를 복원하는 p함수입니다. q함수에선 이미지에 노이즈를 생성하는 간단한 작업을 하지만 이 과정에선 원본 복원을 위해 많은 가중치를 주고 학습합니다. 엄청난 양의 그림과 텍스트 데이터쌍을 사용해 계속 학습시키며 이 노이즈를 복원하면 어떠한 그림이 된다 하는 과정을 계속 반복합니다.
* 앞서 말씀드린 노이즈화와 데이터화의 과정을 실제 고양이 사진을 이용한 예시입니다.
* 다시 한 번 요약하면 노이즈가 가득한 상태의 랜덤 이미지 텐서에서 이미지 디코딩을 통해 노이즈를 점점 제거하며 사용자가 희망하는 그림과 유사한 결과물을 얻을 수 있습니다.
* 다음은 코드 설명입니다. 사용한 라이브러리들은 다음과 같습니다. 이미지를 뽑아내는데 시간이 얼마나 걸렸는지 측정하는 time, 스테이블 디퓨전 모델의 파이프라인으로 사용하는 케라스씨브이는 가장 성능이 좋아 이미지를 뽑아내는데까지 가장 적은 시간이 소모된다는 분석 결과가 있어 사용하게 되었습니다. 그라디오는 머신 러닝을 앱 형태로 실행을 가능하게 만들어줍니다. 트랜슬래이터는 한국어로 입력해도 이미지를 뽑아낼 수 있도록 입력한 키워드를 영어로 번역해주는 역할을 합니다. 우선 케라스 모델에 이미지를 가로 세로 512 픽셀의 크기로 호출하였습니다.
* 왼쪽 플롯 이미지 함수는 맷플롭립을 사용하여 이미지 데이터를 시각화하는데 사용합니다. 오른쪽의 제너레이트 이미지 함수는 입력한 키워드를 받아 영어로 번역하고 모델에 넣어 텍스트를 이미지로 변환하는 역할을 수행합니다.
* 앞서 말씀드린 것처럼 그라디오를 통해 인풋으로 텍스트가 들어오면 아웃풋으로 이미지가 나가게 하고 실행합니다. 실제 수행 화면은 오른쪽과 같습니다.
* 이제 코드 수행 결과를 보여드리고자 합니다. 세 그림은 제가 임의의 키워드들을 넣었을 때 뚜렷하게 나왔거나 개인적으로 마음에 들었던 결과물들을 추려보았습니다. 입력한 키워드들은 아래 적혀있는 것과 같습니다. 한국어로 입력을 했고 트랜슬레이터 라이브러리를 통해 자동으로 번역 이 된 문장이 그 아래 적혀있습니다.
* 남극에 사는 코끼리라는 키워드를 넣었을 때 화면입니다. 이미지를 출력하는데까지 약 30초에 가까운 시간이 소요돼 편집하고 결과물만 보여드렸습니다. 제법 그럴듯한 결과물이 나오는 것을 확인할 수 있습니다.
* 다음 수행 결과입니다. 스테이블 디퓨전 모델이 생각하는 세상에서 가장 귀여운 고양이 이미지를 출력해보겠습니다.
* 마지막으로 서울 밤의 야경이라는 키워드입니다.
* 이상으로 발표를 마치겠습니다. 감사합니다.