Protopia 이미지 변환을 통한 책임 있는 AI

NetApp artificial intelligence solutions

NetApp
August 18, 2025

This PDF was generated from https://docs.netapp.com/ko-kr/netapp-solutions-ai/software/ai-conf-protopia-overview.html on August 18, 2025. Always check docs.netapp.com for the latest.

목차

Protopia 이미지 변환을 통한 책임 있는 AI . . . 1
TR-4928: 책임 있는 AI 및 기밀 추론 - Protopia 이미지 및 데이터 변환을 갖춘 NetApp AI . . . 1
타겟 고객층 . . . 1
솔루션 아키텍처 . . . 2
솔루션 영역 . . . 3
환경 지능 . . . 4
엣지 디바이스 웨어러블 . . . 4
비전투원 대피 작전 . . . 4
의료 및 생물의학 연구 . . . 4
AI/ML 분석의 클라우드 마이그레이션 . . . 4
기술 개요 . . . 4
프로토피아 . . . 5
NetApp ONTAP AI . . . 5
NetApp ONTAP . . . 6
NetApp DataOps 툴킷 . . . 6
NVIDIA Triton 추론 서버 . . . 7
파이토치 . . . 7
NetApp Astra Control . . . 8
NetApp Trident . . . 8
NetApp BlueXP 복사 및 동기화 . . . 8
NetApp BlueXP 분류 . . . 8
테스트 및 검증 계획 . . . 8
테스트 구성 . . . 9
테스트 절차 . . . 9
필수 조건 . . . 9
시나리오 1 – JupyterLab에서의 주문형 추론 . . . 9
시나리오 2 – Kubernetes에서의 일괄 추론 . . . 14
시나리오 3 – NVIDIA Triton 추론 서버 . . . 19
추론 정확도 비교 . . . 23
난독화 속도 . . . 24
결론 . . . 24
추가 정보 및 감사의 말씀을 찾을 수 있는 곳 . . . 25
감사의 말 . . . 26

Protopia 이미지 변환을 통한 책임 있는 AI

TR-4928: 책임 있는 AI 및 기밀 추론 - Protopia 이미지 및 데이터 변환을 갖춘 NetApp AI

Sathish Thyagarajan, Michael Oglesby, NetApp 안병훈, Jennifer Cwagenberg, Protopia

이미지 캡처와 이미지 처리의 등장으로 시각적 해석은 의사소통의 필수적인 부분이 되었습니다. 디지털 영상 처리 분야의 인공지능(AI)은 암 및 기타 질병 식별을 위한 의료 분야, 환경적 위험 연구를 위한 지리공간적 시각 분석, 패턴 인식, 범죄와 싸우기 위한 비디오 처리 등 새로운 사업 기회를 가져다줍니다. 하지만 이러한 기회에는 엄청난 책임도 따릅니다.

조직이 AI에게 더 많은 결정을 맡길수록 데이터 개인정보 보호 및 보안, 법적, 윤리적, 규제적 문제와 관련된 위험을 감수하게 됩니다. 책임 있는 AI는 기업과 정부 기관이 대규모 기업에서 AI를 대규모로 사용하는 데 중요한 신뢰와 거버넌스를 구축할 수 있는 관행을 가능하게 합니다. 이 문서에서는 NetApp 데이터 관리 기술과 Protopia 데이터 난독화 소프트웨어를 사용하여 민감한 데이터를 비공개로 처리하고 위험과 윤리적 문제를 줄이는 세 가지 시나리오에서 NetApp 이 검증한 AI 추론 솔루션을 설명합니다.

소비자와 기업 모두 다양한 디지털 기기를 통해 매일 수백만 개의 이미지를 생성합니다. 이로 인해 데이터와 컴퓨팅 작업 부하가 엄청나게 늘어나면서 기업은 규모와 효율성을 위해 클라우드 컴퓨팅 플랫폼으로 전환하게 되었습니다. 한편, 이미지 데이터에 포함된 민감한 정보에 대한 개인정보 보호 우려가 퍼블릭 클라우드로 전송되면서 발생합니다. 보안 및 개인정보 보호 보장의 부족은 이미지 처리 AI 시스템 구축의 주요 장애물이 됩니다.

또한, "삭제 권리" GDPR에 따르면 개인은 조직에 자신의 모든 개인 데이터를 삭제하도록 요청할 권리가 있습니다. 또한 있습니다 "개인정보보호법" 공정한 정보 관행에 대한 규정을 제정한 법률입니다. 사진과 같은 디지털 이미지는 GDPR에 따라 개인 데이터로 간주될 수 있습니다. GDPR은 데이터를 수집, 처리, 삭제하는 방법을 규정합니다. 이를 이행하지 않을 경우 GDPR을 준수하지 않는 것으로 간주되어, 규정 위반에 대한 엄청난 벌금이 부과될 수 있으며, 이는 조직에 심각한 피해를 줄 수 있습니다. 개인정보 보호 원칙은 머신 러닝(ML) 및 딥 러닝(DL) 모델 예측의 공정성을 보장하고 개인정보 보호 또는 규정 준수 위반과 관련된 위험을 낮추는 책임 있는 AI 구현의 핵심입니다.

이 문서에서는 개인 정보 보호 및 책임 있는 AI 솔루션 배포와 관련된 이미지 난독화가 있는 세 가지 시나리오에서 검증된 설계 솔루션을 설명합니다.

- 시나리오 **1.** Jupyter Notebook에서 주문형 추론이 가능합니다.
- 시나리오 **2.** Kubernetes에서의 일괄 추론.
- 시나리오 **3.** NVIDIA Triton 추론 서버.

이 솔루션을 위해, 우리는 제약 없는 얼굴 감지 문제를 연구하기 위해 설계된 얼굴 영역의 데이터 세트인 FDDB(Face Detection Data Set and Benchmark)를 사용하는데, 이는 FaceBox 구현을 위한 PyTorch 머신 러닝 프레임워크와 결합되었습니다. 이 데이터 세트에는 다양한 해상도의 2845개 이미지 세트에 있는 5171개 얼굴에 대한 주석이 포함되어 있습니다. 또한 이 기술 보고서는 NetApp 고객과 현장 엔지니어로부터 이 솔루션을 적용할 수 있는 상황에서 수집한 일부 솔루션 영역과 관련 사용 사례를 제시합니다.

타겟 고객층

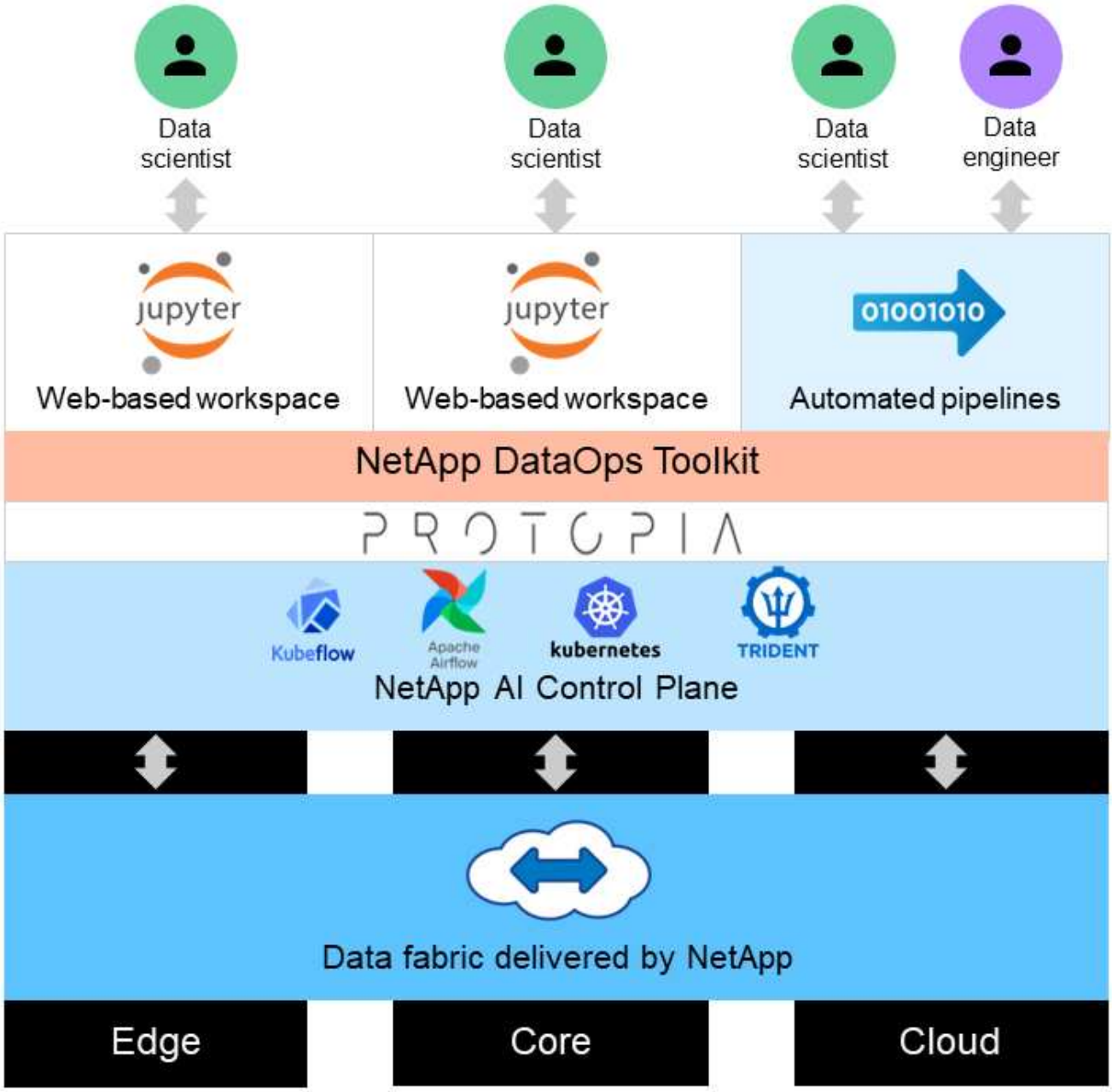
이 기술 보고서는 다음 독자를 대상으로 합니다.

- 공공 장소에서 얼굴 이미지 처리와 관련된 데이터 보호 및 개인 정보 보호 문제를 해결하고 책임감 있는 AI를 설계하고 배포하고자 하는 기업 리더와 엔터프라이즈 설계자.

- 개인정보를 보호하고 보존하는 것을 목표로 하는 데이터 과학자, 데이터 엔지니어, AI/머신러닝(ML) 연구자, AI/ML 시스템 개발자.
- GDPR, CCPA, 국방부(DoD) 및 정부 기관의 개인정보 보호법 등의 규제 표준을 준수하는 AI/ML 모델 및 애플리케이션을 위한 데이터 난독화 솔루션을 설계하는 엔터프라이즈 아키텍트입니다.
- 민감한 정보를 보호하는 딥 러닝(DL) 및 AI/ML/DL 추론 모델을 배포하는 효율적인 방법을 찾고 있는 데이터 과학자와 AI 엔지니어.
- 에지 추론 모델의 배포와 관리를 담당하는 에지 장치 관리자와 에지 서버 관리자입니다.

솔루션 아키텍처

이 솔루션은 기존 CPU와 함께 GPU의 처리 능력을 활용하여 대규모 데이터 세트에 대한 실시간 및 일괄 추론 AI 워크로드를 처리하도록 설계되었습니다. 이 검증은 책임 있는 AI 배포를 추구하는 조직에 필요한 ML의 개인 정보 보호 추론과 최적의 데이터 관리를 보여줍니다. 이 솔루션은 Jupyter Lab 및 CLI 인터페이스를 사용하여 NetApp ONTAP AI(온프레미스 핵심), NetApp DataOps Toolkit, Protopia 난독화 소프트웨어와 상호 연결된 엣지 및 클라우드 컴퓨팅을 위한 단일 또는 다중 노드 Kubernetes 플랫폼에 적합한 아키텍처를 제공합니다. 다음 그림은 NetApp 과 DataOps Toolkit 및 Protopia가 지원하는 데이터 패브릭의 논리적 아키텍처 개요를 보여줍니다.



Protopia 난독화 소프트웨어는 NetApp DataOps Toolkit 상에서 원활하게 실행되며 스토리지 서버를 떠나기 전에 데이터를 변환합니다.

솔루션 영역

디지털 이미지 처리에는 많은 장점이 있으며, 이를 통해 많은 조직이 시각적 표현과 관련된 데이터를 최대한 활용할 수 있습니다. NetApp 과 Protopia 솔루션은 ML/DL 수명 주기 전반에 걸쳐 AI/ML 데이터를 보호하고 비공개화하는 고유한 AI 추론 설계를 제공합니다. 이를 통해 고객은 민감한 데이터의 소유권을 유지하고, 개인정보 보호와 관련된 우려를 완화하여 규모와 효율성을 위해 퍼블릭 또는 하이브리드 클라우드 배포 모델을 사용하고, 엣지에서 AI 추론을 배포할 수 있습니다.

환경 지능

산업계에서는 환경적 위험 측면에서 지리공간 분석을 활용할 수 있는 다양한 방법이 있습니다. 정부와 공공사업부는 전염병이나 산불과 같은 자연재해 발생 시 대중에게 더 나은 조언을 제공하기 위해 공중 보건과 기상 상황에 대한 실행 가능한 통찰력을 얻을 수 있습니다. 예를 들어, 공항이나 병원과 같은 공공 장소에서 영향을 받은 개인의 사생활을 침해하지 않고 COVID-19 양성 환자를 식별하고 해당 당국과 주변 대중에게 필요한 안전 조치를 알릴 수 있습니다.

엣지 디바이스 웨어러블

군대와 전장에서 최첨단 AI 추론을 웨어러블 기기로 사용하여 군인의 건강을 추적하고, 운전자의 행동을 모니터링하고, 군용 차량에 접근하는 것과 관련된 안전 및 위험에 대해 당국에 경고하는 동시에 군인의 개인 정보를 보호하고 보존할 수 있습니다. 군대의 미래는 전장 사물 인터넷(IoBT)과 군사 사물 인터넷(IoMT)을 통해 첨단 기술로 전환되고 있으며, 이는 빠른 엣지 컴퓨팅을 사용하여 군인들이 적을 식별하고 전투에서 더 나은 성과를 낼 수 있도록 돕는 착용형 전투 장비를 의미합니다. 드론이나 웨어러블 장비와 같은 엣지 장치에서 수집된 시각적 데이터를 보호하고 보존하는 것은 해커와 적을 멀리하는 데 매우 중요합니다.

비전투원 대피 작전

비전투원 대피 작전(NEO)은 국방부가 수행하며, 생명이 위험한 미국 시민과 국민, 국방부 민간인, 지정된 사람(주재국(HN)과 제3국 국민(TCN))을 적절한 안전한 피난처로 대피시키는 것을 돕기 위해 수행됩니다. 시행 중인 행정 통제는 대부분 수동 대피자 심사 절차를 사용합니다. 그러나 대피자 식별, 대피자 추적, 위협 스크리닝의 정확성, 보안, 속도는 AI/ML 비디오 난독화 기술과 결합된 고도로 자동화된 AI/ML 도구를 사용하면 잠재적으로 개선될 수 있습니다.

의료 및 생물의학 연구

영상 처리란 컴퓨터 단층촬영(CT)이나 자기공명영상(MRI)으로부터 얻은 3D 영상을 바탕으로 수술 계획을 위한 병리학적 진단을 내리는 데 사용됩니다. HIPAA 개인정보 보호 규칙은 조직이 모든 개인정보 및 사진과 같은 디지털 이미지에 대한 데이터를 수집, 처리 및 삭제하는 방법을 규정합니다. HIPAA 안전 항구 규정에 따라 데이터가 공유 가능한 것으로 간주되려면 얼굴 전체가 나온 사진 이미지와 이와 비슷한 이미지를 삭제해야 합니다. 개인의 얼굴 특징을 구조적 CT/MR 이미지에서 가리는 데 사용되는 익명화나 두개골 제거 알고리즘과 같은 자동화된 기술은 생물의학 연구 기관의 데이터 공유 프로세스에 필수적인 부분이 되었습니다.

AI/ML 분석의 클라우드 마이그레이션

기업 고객은 전통적으로 온프레미스에서 AI/ML 모델을 훈련하고 배포해 왔습니다. 규모의 경제와 효율성을 이유로 이러한 고객은 AI/ML 기능을 퍼블릭, 하이브리드 또는 멀티 클라우드 배포로 옮기는 방향으로 확장하고 있습니다. 그러나 다른 인프라에 노출될 수 있는 데이터에 따라 제한을 받습니다. NetApp 솔루션은 필요한 모든 사이버 보안 위협을 해결합니다. "데이터 보호" 보안 평가를 수행하고 Protopia 데이터 변환과 결합하면 이미지 처리 AI/ML 워크로드를 클라우드로 마이그레이션하는 데 따른 위험을 최소화할 수 있습니다.

다른 산업 분야에서 엣지 컴퓨팅 및 AI 추론에 대한 추가 사용 사례는 다음을 참조하세요."TR-4886 엣지에서의 AI 추론" 그리고 NetApp AI 블로그, "지능 대 개인 정보 보호" .

기술 개요

이 섹션에서는 이 솔루션을 완성하는 데 필요한 다양한 기술 구성 요소에 대한 개요를 제공합니다.

프로토피아

Protopia AI는 현재 시장에서 기밀 추론을 위한 눈에 띄지 않는 소프트웨어 전용 솔루션을 제공합니다. Protopia 솔루션은 민감한 정보의 노출을 최소화하여 추론 서비스에 대한 탁월한 보호 기능을 제공합니다. AI는 현재 작업을 수행하는 데 정말로 필수적인 데이터 레코드에 있는 정보만 제공하고 그 이상은 제공하지 않습니다. 대부분의 추론 작업은 모든 데이터 레코드에 존재하는 모든 정보를 사용하지 않습니다. AI가 이미지, 음성, 비디오 또는 구조화된 표 형식 데이터를 사용하는지 여부에 관계없이 Protopia는 추론 서비스에 필요한 것만 제공합니다. 특허받은 핵심 기술은 수학적으로 큐레이팅된 노이즈를 사용하여 데이터를 확률적으로 변환하고 주어진 ML 서비스에 필요하지 않은 정보를 왜곡합니다. 이 솔루션은 데이터를 가리지 않습니다. 오히려 큐레이팅된 무작위 노이즈를 사용하여 데이터 표현을 변경합니다.

Protopia 솔루션은 모델의 기능과 관련하여 입력 피처 공간에서 관련 정보를 유지하는 동시에 그래디언트 기반 섭동 극대화 방법으로 표현을 변경하는 문제를 공식화합니다. 이 발견 과정은 ML 모델 학습이 끝난 후 미세 조정 단계로 실행됩니다. 패스가 자동으로 일련의 확률 분포를 생성한 후, 로우 오버헤드 데이터 변환을 통해 이러한 분포의 노이즈 샘플을 데이터에 적용하여 추론을 위해 모델에 전달하기 전에 난독화합니다.

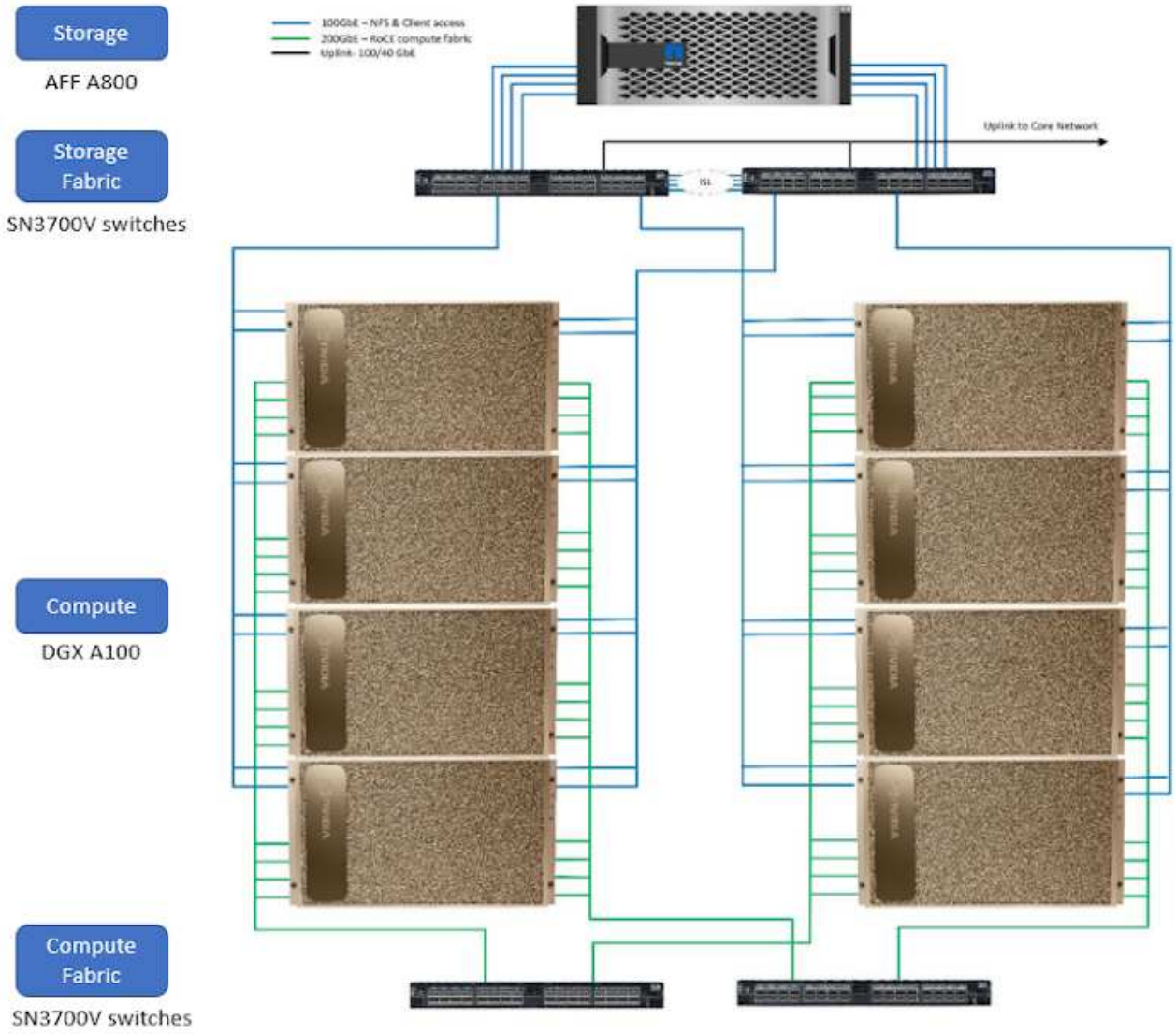
NetApp ONTAP AI

DGX A100 시스템과 NetApp 클라우드 연결 스토리지 시스템으로 구동되는 NetApp ONTAP AI 참조 아키텍처는 NetApp 과 NVIDIA 에서 개발 및 검증했습니다. IT 조직에 다음과 같은 이점을 제공하는 아키텍처를 제공합니다.

- 디자인의 복잡성을 제거합니다
- 컴퓨팅 및 스토리지의 독립적인 확장을 허용합니다.
- 고객이 소규모로 시작하여 원활하게 확장할 수 있도록 지원합니다.
- 다양한 성능 및 비용 지점에 맞는 다양한 스토리지 옵션을 제공합니다.

ONTAP AI는 DGX A100 시스템과 NetApp AFF A800 스토리지 시스템을 최첨단 네트워킹과 긴밀하게 통합합니다. ONTAP AI는 설계의 복잡성과 추측을 제거하여 AI 배포를 간소화합니다. 고객은 소규모로 시작하여 중단 없이 확장할 수 있으며, 엣지에서 코어, 클라우드로 데이터를 지능적으로 관리할 수 있습니다.

다음 그림은 DGX A100 시스템을 기반으로 한 ONTAP AI 솔루션 제품군의 여러 변형을 보여줍니다. AFF A800 시스템 성능은 최대 8개의 DGX A100 시스템으로 검증되었습니다. ONTAP 클러스터에 스토리지 컨트롤러 쌍을 추가하면 아키텍처가 여러 랙으로 확장되어 선형적 성능으로 많은 DGX A100 시스템과 페타바이트 규모의 스토리지 용량을 지원할 수 있습니다. 이 접근 방식은 사용되는 DL 모델의 크기와 필요한 성능 지표에 따라 컴퓨팅 대 스토리지 비율을 독립적으로 변경할 수 있는 유연성을 제공합니다.



ONTAP AI에 대한 추가 정보는 다음을 참조하세요. "NVA-1153: NVIDIA DGX A100 시스템과 Mellanox Spectrum 이더넷 스위치를 탑재한 NetApp ONTAP AI."

NetApp ONTAP

NetApp 의 최신 스토리지 관리 소프트웨어 ONTAP 9.11을 사용하면 기업이 인프라를 현대화하고 클라우드 지원 데이터 센터로 전환할 수 있습니다. ONTAP 업계 최고의 데이터 관리 역량을 활용하여 데이터가 어디에 있든 단일 도구 세트를 사용하여 데이터를 관리하고 보호할 수 있도록 지원합니다. 또한 필요한 곳, 즉 엣지, 코어, 클라우드로 데이터를 자유롭게 이동할 수 있습니다. ONTAP 9.11에는 데이터 관리를 간소화하고, 중요 데이터를 가속화하고 보호하며, 하이브리드 클라우드 아키텍처 전반에서 차세대 인프라 기능을 구현하는 다양한 기능이 포함되어 있습니다.

NetApp DataOps 툴킷

NetApp DataOps Toolkit은 개발자, 데이터 과학자, DevOps 엔지니어, 데이터 엔지니어가 새로운 데이터 볼륨이나 JupyterLab 작업 공간의 거의 즉각적인 프로비저닝, 데이터 볼륨이나 JupyterLab 작업 공간의 거의 즉각적인 복제, 추적성이나 기준 설정을 위한 데이터 볼륨이나 JupyterLab 작업 공간의 거의 즉각적인 스냅샷 촬영 등 다양한 데이터 관리 작업을 간편하게 수행할 수 있도록 해주는 Python 라이브러리입니다. 이 Python 라이브러리는 명령줄 유틸리티로 작동할 수도 있고, 모든 Python 프로그램이나 Jupyter Notebook으로 가져올 수 있는 함수 라이브러리로 작동할 수도 있습니다.

NVIDIA Triton 추론 서버

NVIDIA Triton Inference Server는 프로덕션에서 빠르고 확장 가능한 AI를 제공하기 위해 모델 배포 및 실행을 표준화하는 데 도움이 되는 오픈 소스 추론 제공 소프트웨어입니다. Triton Inference Server는 팀이 GPU 또는 CPU 기반 인프라의 모든 프레임워크에서 학습된 AI 모델을 배포, 실행 및 확장할 수 있도록 하여 AI 추론을 간소화합니다. Triton Inference Server는 TensorFlow, NVIDIA TensorRT, PyTorch, MXNet, OpenVINO 등 모든 주요 프레임워크를 지원합니다. Triton은 모든 주요 퍼블릭 클라우드 AI 및 Kubernetes 플랫폼에서 사용할 수 있는 오케스트레이션 및 확장을 위해 Kubernetes와 통합됩니다. 또한 다양한 MLOps 소프트웨어 솔루션과 통합되어 있습니다.

파이토치

"파이토치"오픈소스 ML 프레임워크입니다. GPU와 CPU를 사용하는 딥러닝을 위한 최적화된 텐서 라이브러리입니다. PyTorch 패키지에는 다차원 텐서에 대한 데이터 구조가 포함되어 있으며, 텐서의 효율적인 직렬화를 위한 여러 유틸리티를 비롯한 다양한 유용한 유틸리티를 제공합니다. 또한 NVIDIA GPU에서 컴퓨팅 기능을 사용하여 텐서 계산을 실행할 수 있는 CUDA 대응 기능도 있습니다. 이 검증에서는 OpenCV-Python(cv2) 라이브러리를 사용하여 Python의 가장 직관적인 컴퓨터 비전 개념을 활용하면서 모델을 검증합니다.

데이터 관리 간소화

적절한 리소스가 AI 애플리케이션과 AI/ML 데이터 세트 교육에 사용될 수 있도록 기업 IT 운영과 데이터 과학자에게 데이터 관리가 매우 중요합니다. NetApp 기술에 대한 다음 추가 정보는 이 검증 범위를 벗어나지만 배포에 따라 관련이 있을 수 있습니다.

ONTAP 데이터 관리 소프트웨어에는 다음과 같은 기능이 포함되어 있어 운영을 간소화하고 단순화하며 총 운영 비용을 절감할 수 있습니다.

- 인라인 데이터 압축 및 확장된 중복 제거. 데이터 압축은 저장 블록 내부의 낭비되는 공간을 줄이고, 중복 제거는 효과적인 용량을 크게 증가시킵니다. 이는 로컬에 저장된 데이터와 클라우드에 계층화된 데이터 모두에 적용됩니다.
- 최소, 최대 및 적응형 서비스 품질(AQoS). 세분화된 서비스 품질(QoS) 제어는 공유 빈도가 높은 환경에서 중요한 애플리케이션의 성능 수준을 유지하는 데 도움이 됩니다.
- NetApp FabricPool. Amazon Web Services(AWS), Azure, NetApp StorageGRID 스토리지 솔루션을 포함한 퍼블릭 및 프라이빗 클라우드 스토리지 옵션에 콜드 데이터의 자동 계층화를 제공합니다. FabricPool 에 대한 자세한 내용은 다음을 참조하세요. "TR-4598: FabricPool 모범 사례" .

데이터 가속화 및 보호

ONTAP 뛰어난 수준의 성능과 데이터 보호 기능을 제공하며 다음과 같은 방식으로 이러한 기능을 확장합니다.

- 성능과 낮은 지연 시간. ONTAP 가능한 가장 낮은 지연 시간으로 가능한 가장 높은 처리량을 제공합니다.
- 데이터 보호. ONTAP 모든 플랫폼에서 공통적으로 관리할 수 있는 내장형 데이터 보호 기능을 제공합니다.
- NetApp 볼륨 암호화(NVE). ONTAP 온보드 및 외부 키 관리 지원을 통해 기본 볼륨 수준 암호화를 제공합니다.
- 다중 테넌시 및 다중 요소 인증. ONTAP 최고 수준의 보안을 통해 인프라 리소스를 공유할 수 있도록 합니다.

미래 지향적 인프라

ONTAP 다음과 같은 기능을 통해 까다롭고 끊임없이 변화하는 비즈니스 요구 사항을 충족하는 데 도움이 됩니다.

- 원활한 확장과 중단 없는 운영. ONTAP 기존 컨트롤러와 확장형 클러스터에 중단 없이 용량을 추가할 수 있도록 지원합니다. 고객은 비용이 많이 드는 데이터 마이그레이션이나 중단 없이 NVMe 및 32Gb FC와 같은 최신 기술로

업그레이드할 수 있습니다.

- 클라우드 연결. ONTAP 모든 퍼블릭 클라우드에서 소프트웨어 정의 스토리지(ONTAP Select)와 클라우드 기반 인스턴스(Google Cloud NetApp Volumes)에 대한 옵션을 제공하는 가장 클라우드에 연결된 스토리지 관리 소프트웨어입니다.
- 새로운 애플리케이션과의 통합. ONTAP 기존 엔터프라이즈 앱을 지원하는 동일한 인프라를 사용하여 자율주행차, 스마트 시티, 산업 4.0과 같은 차세대 플랫폼과 애플리케이션을 위한 엔터프라이즈급 데이터 서비스를 제공합니다.

NetApp Astra Control

NetApp Astra 제품군은 NetApp 스토리지 및 데이터 관리 기술을 기반으로 온프레미스와 퍼블릭 클라우드에서 Kubernetes 애플리케이션을 위한 스토리지 및 애플리케이션 인식 데이터 관리 서비스를 제공합니다. Kubernetes 애플리케이션을 쉽게 백업하고, 다른 클러스터로 데이터를 마이그레이션하고, 작동하는 애플리케이션 복제본을 즉시 생성할 수 있습니다. 퍼블릭 클라우드에서 실행되는 Kubernetes 애플리케이션을 관리해야 하는 경우 다음 문서를 참조하세요. "Astra 컨트롤 서비스" . Astra Control Service는 NetApp 에서 관리하는 서비스로, Google Kubernetes Engine(GKE)과 Azure Kubernetes Service(AKS)에서 Kubernetes 클러스터의 애플리케이션 인식 데이터 관리를 제공합니다.

NetApp Trident

Astra "Trident" NetApp 의 Docker와 Kubernetes를 위한 오픈소스 동적 스토리지 오케스트레이터로, 영구 스토리지의 생성, 관리 및 사용을 간소화합니다. Kubernetes 기반 애플리케이션인 Trident 는 Kubernetes 클러스터 내에서 직접 실행됩니다. Trident 사용하면 고객이 DL 컨테이너 이미지를 NetApp 스토리지에 원활하게 배포할 수 있으며 AI 컨테이너 배포를 위한 엔터프라이즈급 환경을 제공합니다. Kubernetes 사용자(ML 개발자, 데이터 과학자 등)는 NetApp 기술이 제공하는 고급 데이터 관리 기능을 활용하여 오케스트레이션 및 복제를 생성, 관리 및 자동화할 수 있습니다.

NetApp BlueXP 복사 및 동기화

"BlueXP 복사 및 동기화"빠르고 안전한 데이터 동기화를 위한 NetApp 서비스입니다. 온프레미스 NFS 또는 SMB 파일 공유, NetApp StorageGRID, NetApp ONTAP S3, Google Cloud NetApp Volumes, Azure NetApp Files, Amazon Simple Storage Service(Amazon S3), Amazon Elastic File System(Amazon EFS), Azure Blob, Google Cloud Storage 또는 IBM Cloud Object Storage 간에 파일을 전송해야 하는 경우 BlueXP Copy and Sync를 사용하면 파일을 필요한 곳으로 빠르고 안전하게 이동할 수 있습니다. 데이터가 전송되면 소스와 타겟 모두에서 자유롭게 사용할 수 있습니다. BlueXP Copy and Syncc는 사전 정의된 일정에 따라 데이터를 지속적으로 동기화하여 델타만 이동하므로 데이터 복제에 소요되는 시간과 비용이 최소화됩니다. BlueXP Copy and Sync는 설정과 사용이 매우 간단한 SaaS(소프트웨어 즉 서비스) 도구입니다. BlueXP Copy and Sync에 의해 트리거되는 데이터 전송은 데이터 브로커를 통해 수행됩니다. AWS, Azure, Google Cloud Platform 또는 온프레미스에 BlueXP 복사 및 동기화 데이터 브로커를 배포할 수 있습니다.

NetApp BlueXP 분류

강력한 AI 알고리즘으로 구동됩니다. "NetApp BlueXP 분류" 전체 데이터 자산에 걸쳐 자동화된 제어와 데이터 거버넌스를 제공합니다. 비용 절감 방안을 쉽게 찾고, 규정 준수 및 개인정보 보호 문제를 파악하고, 최적화 기회를 찾을 수 있습니다. BlueXP 분류 대시보드를 사용하면 중복 데이터를 식별하여 중복을 제거하고, 개인 데이터, 비개인 데이터, 민감한 데이터를 매핑하고, 민감한 데이터와 이상 현상에 대한 알림을 켤 수 있는 통찰력을 얻을 수 있습니다.

테스트 및 검증 계획

이 솔루션 설계를 위해 다음 세 가지 시나리오가 검증되었습니다.

- Kubernetes용 NetApp DataOps Toolkit을 사용하여 조율된 JupyterLab 작업 공간 내에서 Protopia 난독화가 적용된 추론 작업과 적용되지 않은 추론 작업입니다.
- Kubernetes에서 NetApp DataOps Toolkit for Kubernetes를 사용하여 데이터 볼륨을 조정한 Protopia 난독화를 적용한 배치 추론 작업과 적용하지 않은 배치 추론 작업입니다.
- Kubernetes용 NetApp DataOps Toolkit을 사용하여 조율된 NVIDIA Triton Inference Server 인스턴스를 사용하는 추론 작업입니다. 네트워크를 통해 전송되는 모든 데이터는 난독화되어야 한다는 일반적인 요구 사항을 시뮬레이션하기 위해 Triton 추론 API를 호출하기 전에 이미지에 Protopia 난독화를 적용했습니다. 이 워크플로는 신뢰할 수 있는 영역 내에서 데이터를 수집했지만 추론을 위해 해당 신뢰할 수 있는 영역 외부로 데이터를 전달해야 하는 사용 사례에 적용할 수 있습니다. Protopia 난독화 없이는 민감한 데이터가 신뢰 영역을 벗어나지 않고 이러한 유형의 워크플로를 구현하는 것은 불가능합니다.

테스트 구성

다음 표는 솔루션 설계 검증 환경을 간략하게 설명합니다.

요소	버전
쿠버네티스	1.21.6
NetApp Trident CSI 드라이버	22.01.0
Kubernetes용 NetApp DataOps 툴킷	2.3.0
NVIDIA Triton 추론 서버	21.11-파이3

테스트 절차

이 섹션에서는 검증을 완료하는 데 필요한 작업을 설명합니다.

필수 조건

이 섹션에 설명된 작업을 실행하려면 다음 도구가 설치 및 구성된 Linux 또는 macOS 호스트에 액세스할 수 있어야 합니다.

- Kubectl(기존 Kubernetes 클러스터에 액세스하도록 구성됨)
  - 설치 및 구성 지침을 찾을 수 있습니다. "여기" .
- Kubernetes용 NetApp DataOps 툴킷
  - 설치 지침을 찾을 수 있습니다 "여기" .

시나리오 **1 – JupyterLab**에서의 주문형 추론

1. AI/ML 추론 워크로드를 위한 Kubernetes 네임스페이스를 만듭니다.

```
$ kubectl create namespace inference
namespace/inference created
```

2. NetApp DataOps Toolkit을 사용하여 추론을 수행할 데이터를 저장할 영구 볼륨을 프로비저닝합니다.

```
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
-name=inference-data --size=50Gi
Creating PersistentVolumeClaim (PVC) 'inference-data' in namespace
'inference'.
PersistentVolumeClaim (PVC) 'inference-data' created. Waiting for
Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'inference-data' in namespace 'inference'.
```

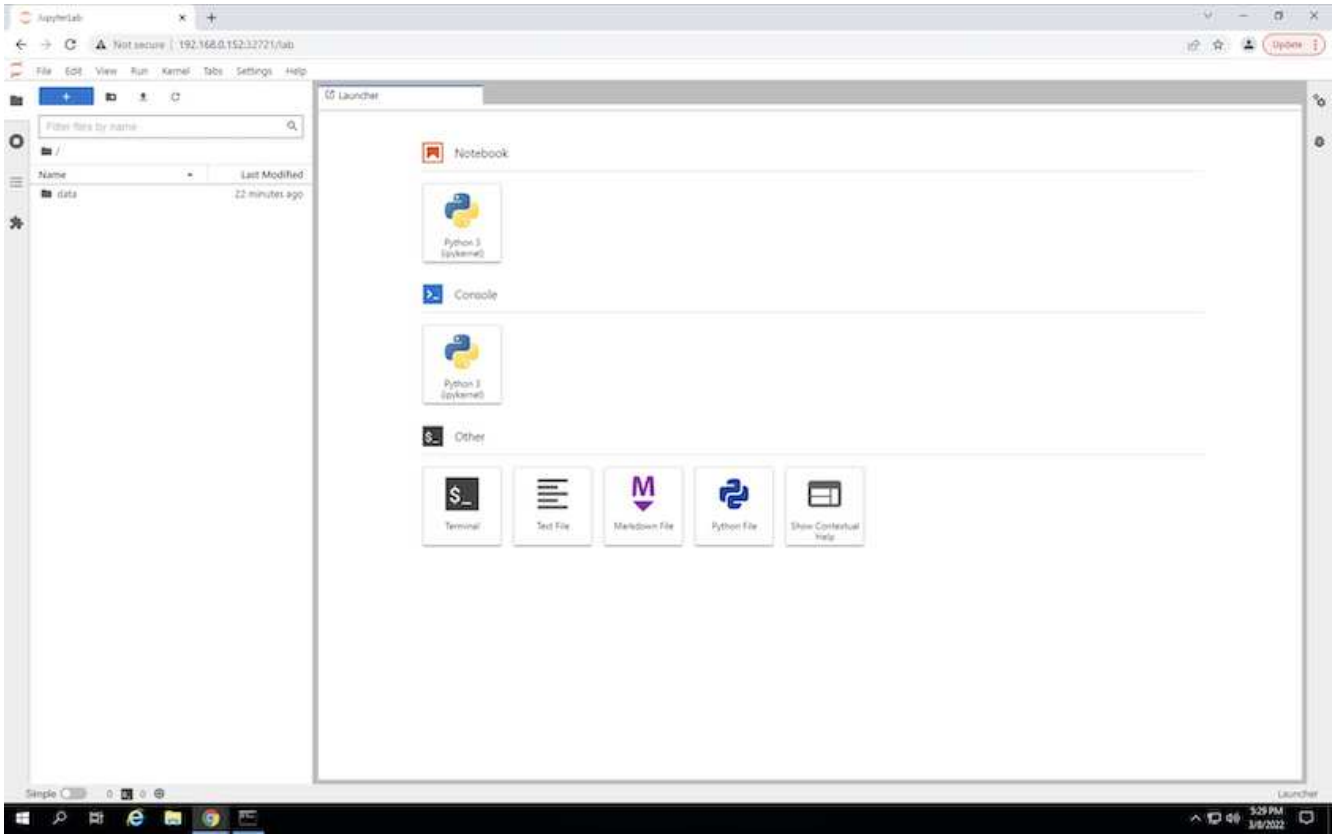
3. NetApp DataOps Toolkit을 사용하여 새로운 JupyterLab 작업 공간을 만듭니다. 이전 단계에서 생성된 영구 볼륨을 다음을 사용하여 마운트합니다. `--mount- pvc` 옵션. 필요에 따라 NVIDIA GPU를 작업 공간에 할당하려면 다음을 사용하십시오. `-- nvidia-gpu` 옵션.

   다음 예에서는 영구 볼륨 `inference-data` JupyterLab 작업 공간 컨테이너에 마운트됩니다. `/home/jovyan/data` . 공식 Project Jupyter 컨테이너 이미지를 사용하는 경우 `/home/jovyan` JupyterLab 웹 인터페이스 내의 최상위 디렉토리로 표시됩니다.

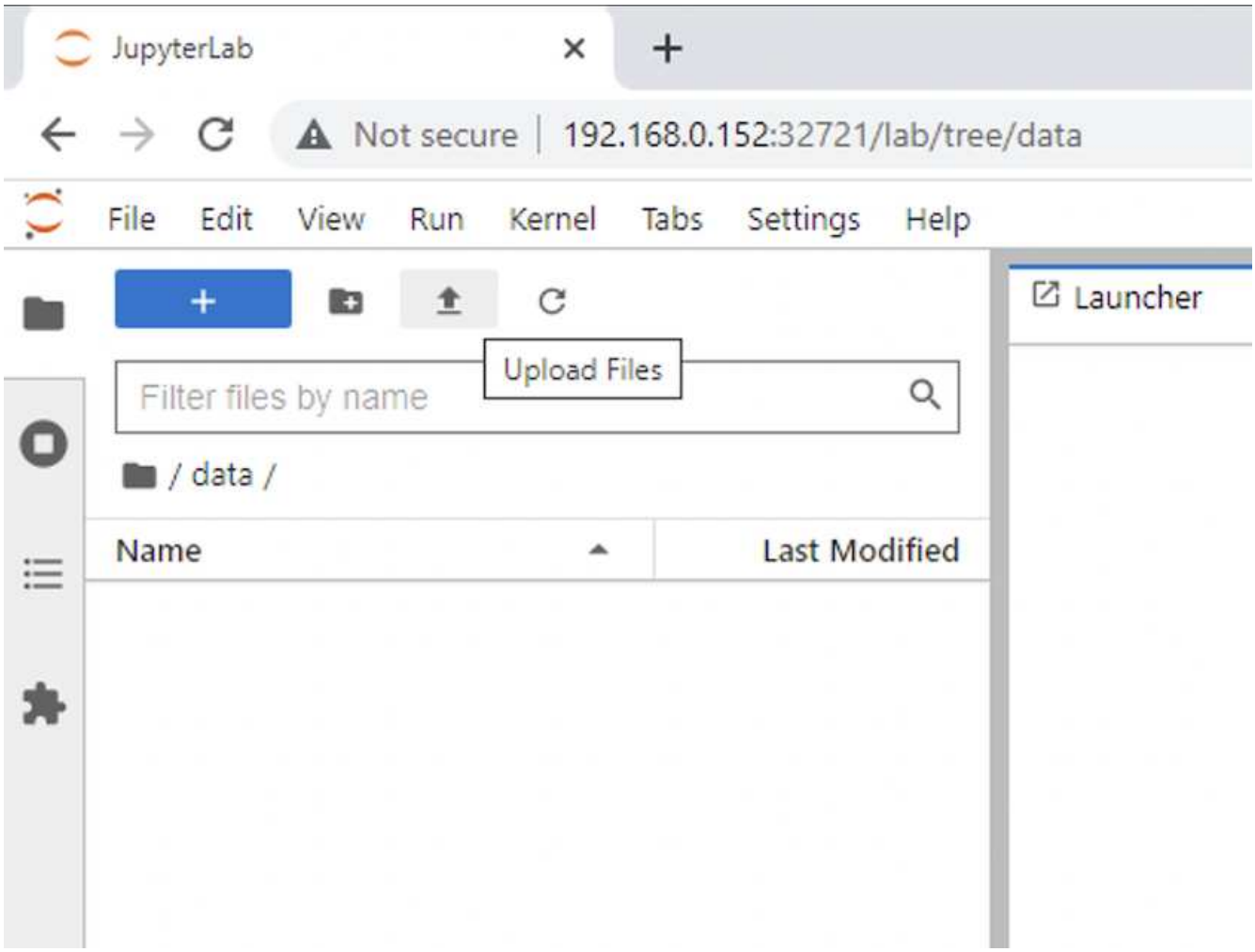
```
$ netapp_dataops_k8s_cli.py create jupyterlab --namespace=inference
--workspace-name=live-inference --size=50Gi --nvidia-gpu=2 --mount
-pvc=inference-data:/home/jovyan/data
Set workspace password (this password will be required in order to
access the workspace):
Re-enter password:
Creating persistent volume for workspace...
Creating PersistentVolumeClaim (PVC) 'ntap-dsutil-jupyterlab-live-
inference' in namespace 'inference'.
PersistentVolumeClaim (PVC) 'ntap-dsutil-jupyterlab-live-inference'
created. Waiting for Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'ntap-dsutil-jupyterlab-live-inference' in namespace 'inference'.
Creating Service 'ntap-dsutil-jupyterlab-live-inference' in namespace
'inference'.
Service successfully created.
Attaching Additional PVC: 'inference-data' at mount_path:
'/home/jovyan/data'.
Creating Deployment 'ntap-dsutil-jupyterlab-live-inference' in namespace
'inference'.
Deployment 'ntap-dsutil-jupyterlab-live-inference' created.
Waiting for Deployment 'ntap-dsutil-jupyterlab-live-inference' to reach
Ready state.
Deployment successfully created.
Workspace successfully created.
To access workspace, navigate to http://192.168.0.152:32721
```

4. 출력에 지정된 URL을 사용하여 JupyterLab 작업 공간에 액세스합니다. `create jupyterlab` 명령. 데이터

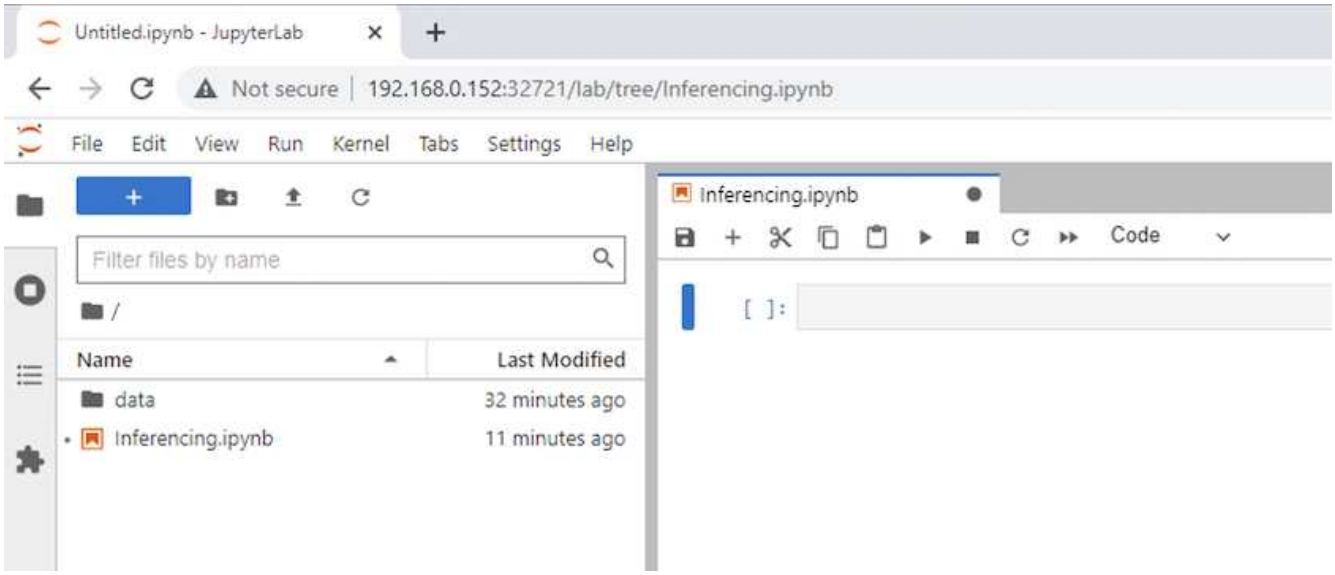
디렉토리는 작업 공간에 마운트된 영구 볼륨을 나타냅니다.



5. 열기 `data` 디렉토리로 가서 추론을 수행할 파일을 업로드합니다. 파일이 데이터 디렉토리에 업로드되면 해당 파일은 작업 공간에 마운트된 영구 볼륨에 자동으로 저장됩니다. 파일을 업로드하려면 다음 이미지에 표시된 대로 파일 업로드 아이콘을 클릭하세요.



6. 최상위 디렉토리로 돌아가서 새로운 노트북을 만듭니다.



7. 노트북에 추론 코드를 추가합니다. 다음 예제는 이미지 감지 사용 사례에 대한 추론 코드를 보여줍니다.

Launcher | image-demo-pytorch.ipynb

Code | Python 3 (ipykernel)

STEP 3-1: Clean (Without obfuscation) detection

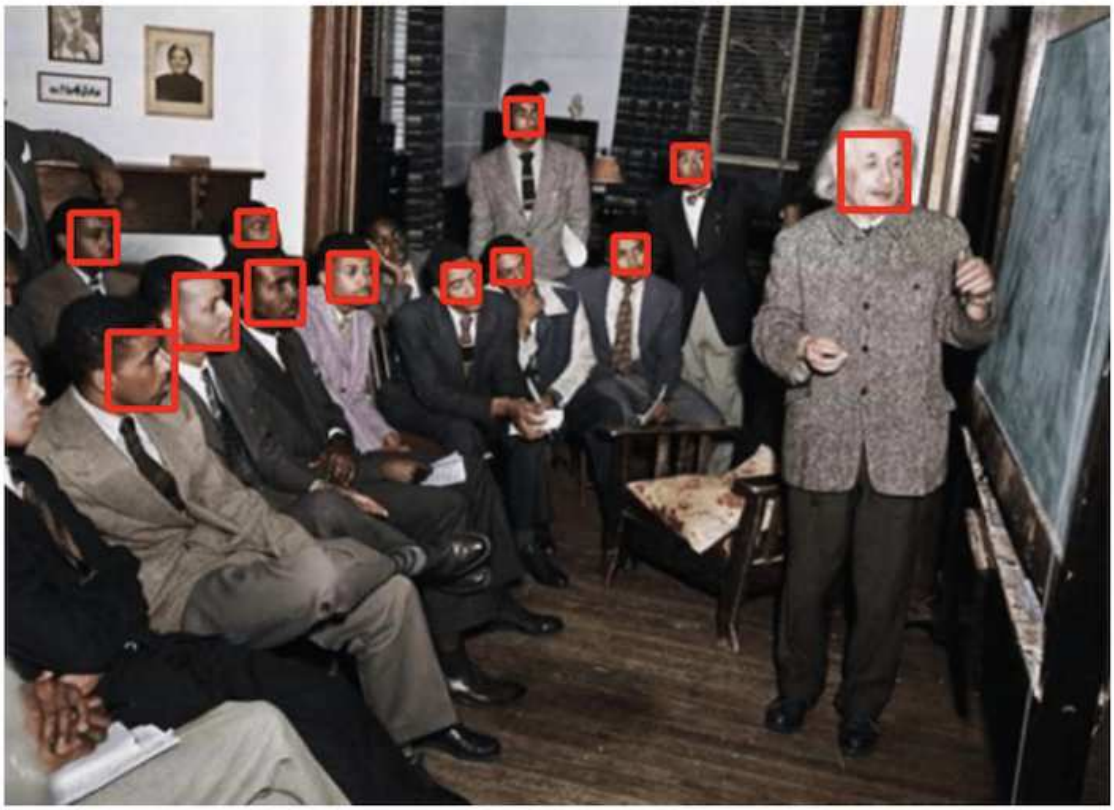
[9]:
```
# get current frame
frame = input_image

# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)

# run forward pass
clean_activation = clean_model.forward_head(preprocessed_input)  # runs the first few layers
loc, pred = clean_model.forward_tail(clean_activation)  # runs rest of the layers

# postprocess output
clean_pred = (loc.detach().cpu().numpy(), pred.detach().cpu().numpy())
clean_outputs = postprocess_outputs(
    clean_pred, [[input_image_width, input_image_height]], priors, THRESHOLD
)

# draw rectangles
clean_frame = copy.deepcopy(frame)  # needs to be deep copy
for (x1, y1, x2, y2, s) in clean_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(clean_frame, (x1, y1), (x2, y2), (0, 0, 255), 4)
```

Launcher | image-demo-pytorch.ipynb

Code | Python 3 (ipykernel)

Visualize Clean (Without obfuscation) detection

[10]:
```
show_cv2_image(clean_frame, scale=2)
```



8. 추론 코드에 Protopia 난독화를 추가하세요. Protopia는 고객과 직접 협력하여 사용 사례별 문서를 제공하며, 이는 본 기술 보고서의 범위를 벗어납니다. 다음 예제에서는 Protopia 난독화가 추가된 이미지 감지 사용 사례에 대한 추론 코드를 보여줍니다.

Launcher | image-demo-pytorch.ipynb

Code | Python 3 (ipykernel)

STEP 3-2: Protopia AI (With obfuscation) detection

[11]:
```
# get current frame
frame = input_image

# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)

# run forward pass
not_noisy_activation = noisy_model.forward_head(preprocessed_input)  # runs the first few layers
##################################################################
#          SINGLE ADITIONAL LINE FOR PRIVATE INFERENCE           #
##################################################################
noisy_activation = noisy_model.forward_noise(not_noisy_activation)
##################################################################
loc, pred = noisy_model.forward_tail(noisy_activation)  # runs rest of the layers

# postprocess output
noisy_pred = (loc.detach().cpu().numpy(), pred.detach().cpu().numpy())
noisy_outputs = postprocess_outputs(
    noisy_pred, [[input_image_width, input_image_height]], priors, THRESHOLD * 0.5
)

# get reconstruction of the noisy activation
noisy_reconstruction = decoder_function(noisy_activation)
noisy_reconstruction = noisy_reconstruction.detach().cpu().numpy()[0]
noisy_reconstruction = unpreprocess_output(
    noisy_reconstruction, (input_image_width, input_image_height), True
).astype(np.uint8)

# draw rectangles
for (x1, y1, x2, y2, s) in noisy_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(noisy_reconstruction, (x1, y1), (x2, y2), (0, 0, 255), 4)
```

Launcher | image-demo-pytorch.ipynb

Code | Python 3 (ipykernel)

Visualize Protopia AI (With obfuscation) detection

[12]:
```
show_cv2_image(noisy_reconstruction, scale=2)
```

시나리오 **2 – Kubernetes**에서의 일괄 추론

1. AI/ML 추론 워크로드를 위한 Kubernetes 네임스페이스를 만듭니다.

```
$ kubectl create namespace inference
namespace/inference created
```

2. NetApp DataOps Toolkit을 사용하여 추론을 수행할 데이터를 저장할 영구 볼륨을 프로비저닝합니다.

```
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
-name=inference-data --size=50Gi
Creating PersistentVolumeClaim (PVC) 'inference-data' in namespace
'inference'.
PersistentVolumeClaim (PVC) 'inference-data' created. Waiting for
Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'inference-data' in namespace 'inference'.
```

3. 추론을 수행할 데이터로 새 영구 볼륨을 채웁니다.

   PVC에 데이터를 로드하는 방법에는 여러 가지가 있습니다. 데이터가 현재 NetApp StorageGRID 또는 Amazon S3와 같은 S3 호환 개체 스토리지 플랫폼에 저장되어 있는 경우 다음을 사용할 수 있습니다. "NetApp DataOps Toolkit S3 Data Mover 기능" . 또 다른 간단한 방법은 JupyterLab 작업 공간을 만든 다음 "3~5단계" 섹션에 설명된 대로 JupyterLab 웹 인터페이스를 통해 파일을 업로드하는 것입니다.시나리오 1 – JupyterLab에서의 주문형 추론 ."

4. 일괄 추론 작업을 위한 Kubernetes 작업을 만듭니다. 다음 예에서는 이미지 감지 사용 사례에 대한 일괄 추론 작업을 보여줍니다. 이 작업은 이미지 세트의 각 이미지에 대한 추론을 수행하고 추론 정확도 측정 항목을 stdout에 기록합니다.

```
$ vi inference-job-raw.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-inference-raw
  namespace: inference
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: inference-data
      - name: dshm
        emptyDir:
          medium: Memory
      containers:
      - name: inference
        image: netapp-protopia-inference:latest
        imagePullPolicy: IfNotPresent
        command: ["python3", "run-accuracy-measurement.py", "--dataset",
"/data/netapp-face-detection/FDDB"]
        resources:
          limits:
            nvidia.com/gpu: 2
        volumeMounts:
        - mountPath: /data
          name: data
        - mountPath: /dev/shm
          name: dshm
      restartPolicy: Never
$ kubectl create -f inference-job-raw.yaml
job.batch/netapp-inference-raw created
```

5. 추론 작업이 성공적으로 완료되었는지 확인하세요.

```
$ kubectl -n inference logs netapp-inference-raw-255sp
100%|██████████| 89/89 [00:52<00:00,  1.68it/s]
Reading Predictions : 100%|██████████| 10/10 [00:01<00:00,  6.23it/s]
Predicting ... : 100%|██████████| 10/10 [00:16<00:00,  1.64s/it]
==================== Results ====================
FDDB-fold-1 Val AP: 0.9491256561145955
FDDB-fold-2 Val AP: 0.9205024466101926
FDDB-fold-3 Val AP: 0.9253013871078468
FDDB-fold-4 Val AP: 0.9399781485863011
FDDB-fold-5 Val AP: 0.9504280149478732
FDDB-fold-6 Val AP: 0.9416473519339292
FDDB-fold-7 Val AP: 0.9241631566241117
FDDB-fold-8 Val AP: 0.9072663297546659
FDDB-fold-9 Val AP: 0.9339648715035469
FDDB-fold-10 Val AP: 0.9447707905560152
FDDB Dataset Average AP: 0.9337148153739079
==================================================
mAP: 0.9337148153739079
```

6. 추론 작업에 Protopia 난독화를 추가하세요. Protopia 난독화를 Protopia에서 직접 추가하는 방법에 대한 사용 사례별 지침은 찾을 수 있지만, 이는 이 기술 보고서의 범위를 벗어납니다. 다음 예제에서는 ALPHA 값 0.8을 사용하여 Protopia 난독화를 추가한 얼굴 감지 사용 사례에 대한 일괄 추론 작업을 보여줍니다. 이 작업은 이미지 세트의 각 이미지에 대한 추론을 수행하기 전에 Protopia 난독화를 적용한 다음 추론 정확도 측정 항목을 stdout에 기록합니다.

   우리는 ALPHA 값 0.05, 0.1, 0.2, 0.4, 0.6, 0.8, 0.9, 0.95에 대해 이 단계를 반복했습니다. 결과는 다음에서 볼 수 있습니다."추론 정확도 비교."

```
$ vi inference-job-protopia-0.8.yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: netapp-inference-protopia-0.8
  namespace: inference
spec:
  backoffLimit: 5
  template:
    spec:
      volumes:
      - name: data
        persistentVolumeClaim:
          claimName: inference-data
      - name: dshm
        emptyDir:
          medium: Memory
      containers:
      - name: inference
        image: netapp-protopia-inference:latest
        imagePullPolicy: IfNotPresent
        env:
        - name: ALPHA
          value: "0.8"
        command: ["python3", "run-accuracy-measurement.py", "--dataset",
"/data/netapp-face-detection/FDDB", "--alpha", "$(ALPHA)", "--noisy"]
        resources:
          limits:
            nvidia.com/gpu: 2
        volumeMounts:
        - mountPath: /data
          name: data
        - mountPath: /dev/shm
          name: dshm
      restartPolicy: Never
$ kubectl create -f inference-job-protopia-0.8.yaml
job.batch/netapp-inference-protopia-0.8 created
```

7. 추론 작업이 성공적으로 완료되었는지 확인하세요.

```
$ kubectl -n inference logs netapp-inference-protopia-0.8-b4dkz
100%|██████████| 89/89 [01:05<00:00,  1.37it/s]
Reading Predictions : 100%|██████████| 10/10 [00:02<00:00,  3.67it/s]
Predicting ... : 100%|██████████| 10/10 [00:22<00:00,  2.24s/it]
==================== Results ====================
FDDB-fold-1 Val AP: 0.8953066115834589
FDDB-fold-2 Val AP: 0.8819580264029936
FDDB-fold-3 Val AP: 0.8781107458462862
FDDB-fold-4 Val AP: 0.9085731346308461
FDDB-fold-5 Val AP: 0.9166445508275378
FDDB-fold-6 Val AP: 0.9101178994188819
FDDB-fold-7 Val AP: 0.8383443678423771
FDDB-fold-8 Val AP: 0.8476311547659464
FDDB-fold-9 Val AP: 0.8739624502111121
FDDB-fold-10 Val AP: 0.8905468076424851
FDDB Dataset Average AP: 0.8841195749171925
==================================================
mAP: 0.8841195749171925
```

시나리오 3 – NVIDIA Triton 추론 서버

1. AI/ML 추론 워크로드를 위한 Kubernetes 네임스페이스를 만듭니다.

```
$ kubectl create namespace inference
namespace/inference created
```

2. NetApp DataOps Toolkit을 사용하여 NVIDIA Triton Inference Server의 모델 저장소로 사용할 영구 볼륨을 프로비저닝합니다.

```
$ netapp_dataops_k8s_cli.py create volume --namespace=inference --pvc
-name=triton-model-repo --size=100Gi
Creating PersistentVolumeClaim (PVC) 'triton-model-repo' in namespace
'inference'.
PersistentVolumeClaim (PVC) 'triton-model-repo' created. Waiting for
Kubernetes to bind volume to PVC.
Volume successfully created and bound to PersistentVolumeClaim (PVC)
'triton-model-repo' in namespace 'inference'.
```

3. 새 영구 볼륨에 모델을 저장합니다. "체재" NVIDIA Triton 추론 서버에서 인식됩니다.

   PVC에 데이터를 로드하는 방법에는 여러 가지가 있습니다. 간단한 방법은 JupyterLab 작업 공간을 만든 다음 "3~5단계"에 설명된 대로 JupyterLab 웹 인터페이스를 통해 파일을 업로드하는 것입니다.시나리오 1 – JupyterLab에서의 주문형 추론 . "

4. NetApp DataOps Toolkit을 사용하여 새로운 NVIDIA Triton Inference Server 인스턴스를 배포합니다.

```
$ netapp_dataops_k8s_cli.py create triton-server --namespace=inference
--server-name=netapp-inference --model-repo-pvc-name=triton-model-repo
Creating Service 'ntap-dsutil-triton-netapp-inference' in namespace
'inference'.
Service successfully created.
Creating Deployment 'ntap-dsutil-triton-netapp-inference' in namespace
'inference'.
Deployment 'ntap-dsutil-triton-netapp-inference' created.
Waiting for Deployment 'ntap-dsutil-triton-netapp-inference' to reach
Ready state.
Deployment successfully created.
Server successfully created.
Server endpoints:
http: 192.168.0.152: 31208
grpc: 192.168.0.152: 32736
metrics: 192.168.0.152: 30009/metrics
```

5. Triton 클라이언트 SDK를 사용하여 추론 작업을 수행합니다. 다음 Python 코드 발췌 부분은 Triton Python 클라이언트 SDK를 사용하여 얼굴 감지 사용 사례에 대한 추론 작업을 수행합니다. 이 예제에서는 Triton API를 호출하고 추론을 위해 이미지를 전달합니다. 그러면 Triton 추론 서버는 요청을 수신하고, 모델을 호출하고, API 결과의 일부로 추론 출력을 반환합니다.

```
# get current frame
frame = input_image
# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)
# run forward pass
clean_activation = clean_model_head(preprocessed_input)  # runs the
first few layers
#######################################################################
##############
#          pass clean image to Triton Inference Server API for
inferencing           #
#######################################################################
##############
triton_client =
httpclient.InferenceServerClient(url="192.168.0.152:31208",
verbose=False)
model_name = "face_detection_base"
inputs = []
outputs = []
inputs.append(httpclient.InferInput("INPUT__0", [1, 128, 32, 32],
```

```
"FP32"))
inputs[0].set_data_from_numpy(clean_activation.detach().cpu().numpy(),
binary_data=False)
outputs.append(httpclient.InferRequestedOutput("OUTPUT__0",
binary_data=False))
outputs.append(httpclient.InferRequestedOutput("OUTPUT__1",
binary_data=False))
results = triton_client.infer(
    model_name,
    inputs,
    outputs=outputs,
    #query_params=query_params,
    headers=None,
    request_compression_algorithm=None,
    response_compression_algorithm=None)
#print(results.get_response())
statistics =
triton_client.get_inference_statistics(model_name=model_name,
headers=None)
print(statistics)
if len(statistics["model_stats"]) != 1:
    print("FAILED: Inference Statistics")
    sys.exit(1)

loc_numpy = results.as_numpy("OUTPUT__0")
pred_numpy = results.as_numpy("OUTPUT__1")
#######################################################################
##############
# postprocess output
clean_pred = (loc_numpy, pred_numpy)
clean_outputs = postprocess_outputs(
    clean_pred, [[input_image_width, input_image_height]], priors,
THRESHOLD
)
# draw rectangles
clean_frame = copy.deepcopy(frame)  # needs to be deep copy
for (x1, y1, x2, y2, s) in clean_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(clean_frame, (x1, y1), (x2, y2), (0, 0, 255), 4)
```

6. 추론 코드에 Protopia 난독화를 추가하세요. Protopia에서 직접 Protopia 난독화를 추가하는 방법에 대한 사용 사례별 지침은 찾을 수 있습니다. 그러나 이 프로세스는 이 기술 보고서의 범위를 벗어납니다. 다음 예제는 이전 단계 5에서 보여준 것과 동일한 Python 코드를 보여주지만, Protopia 난독화가 추가되었습니다.

   Protopia 난독화는 이미지가 Triton API로 전달되기 전에 적용된다는 점에 유의하세요. 따라서 난독화되지 않은

이미지는 로컬 머신을 벗어나지 않습니다. 난독화된 이미지만 네트워크를 통해 전달됩니다. 이 워크플로는 신뢰할 수 있는 영역 내에서 데이터를 수집한 후 추론을 위해 해당 신뢰할 수 있는 영역 외부로 데이터를 전달해야 하는 사용 사례에 적용할 수 있습니다. Protopia 난독화 없이는 민감한 데이터가 신뢰 영역을 벗어나지 않고 이러한 유형의 워크플로를 구현하는 것은 불가능합니다.

```
# get current frame
frame = input_image
# preprocess input
preprocessed_input = preprocess_input(frame)
preprocessed_input = torch.Tensor(preprocessed_input).to(device)
# run forward pass
not_noisy_activation = noisy_model_head(preprocessed_input)  # runs the
first few layers
#################################################################
#          obfuscate image locally prior to inferencing          #
#          SINGLE ADITIONAL LINE FOR PRIVATE INFERENCE           #
#################################################################
noisy_activation = noisy_model_noise(not_noisy_activation)
#################################################################
#######################################################################
####################
#          pass obfuscated image to Triton Inference Server API for
inferencing           #
#######################################################################
####################
triton_client =
httpclient.InferenceServerClient(url="192.168.0.152:31208",
verbose=False)
model_name = "face_detection_noisy"
inputs = []
outputs = []
inputs.append(httpclient.InferInput("INPUT__0", [1, 128, 32, 32],
"FP32"))
inputs[0].set_data_from_numpy(noisy_activation.detach().cpu().numpy(),
binary_data=False)
outputs.append(httpclient.InferRequestedOutput("OUTPUT__0",
binary_data=False))
outputs.append(httpclient.InferRequestedOutput("OUTPUT__1",
binary_data=False))
results = triton_client.infer(
    model_name,
    inputs,
    outputs=outputs,
    #query_params=query_params,
    headers=None,
    request_compression_algorithm=None,
```

```
    response_compression_algorithm=None)
#print(results.get_response())
statistics =
triton_client.get_inference_statistics(model_name=model_name,
headers=None)
print(statistics)
if len(statistics["model_stats"]) != 1:
    print("FAILED: Inference Statistics")
    sys.exit(1)

loc_numpy = results.as_numpy("OUTPUT__0")
pred_numpy = results.as_numpy("OUTPUT__1")
#######################################################################
###################

# postprocess output
noisy_pred = (loc_numpy, pred_numpy)
noisy_outputs = postprocess_outputs(
    noisy_pred, [[input_image_width, input_image_height]], priors,
THRESHOLD * 0.5
)
# get reconstruction of the noisy activation
noisy_reconstruction = decoder_function(noisy_activation)
noisy_reconstruction = noisy_reconstruction.detach().cpu().numpy()[0]
noisy_reconstruction = unpreprocess_output(
    noisy_reconstruction, (input_image_width, input_image_height), True
).astype(np.uint8)
# draw rectangles
for (x1, y1, x2, y2, s) in noisy_outputs[0]:
    x1, y1 = int(x1), int(y1)
    x2, y2 = int(x2), int(y2)
    cv2.rectangle(noisy_reconstruction, (x1, y1), (x2, y2), (0, 0, 255),
4)
```

추론 정확도 비교

이러한 검증을 위해, 우리는 일련의 원시 이미지를 사용하여 이미지 감지 사용 사례에 대한 추론을 수행했습니다. 그런 다음 추론 작업을 하기 전에 Protopia 난독화를 추가한 동일한 이미지 세트에 대해 동일한 추론 작업을 수행했습니다. 우리는 Protopia 난독화 구성 요소에 대해 다른 ALPHA 값을 사용하여 작업을 반복했습니다. Protopia 난독화의 맥락에서 ALPHA 값은 적용되는 난독화 양을 나타내며, ALPHA 값이 높을수록 난독화 수준이 높아짐을 나타냅니다. 그런 다음 우리는 이러한 다양한 실행에 걸쳐 추론 정확도를 비교했습니다.

다음 두 표는 우리의 사용 사례에 대한 세부 정보를 제공하고 결과를 간략하게 설명합니다.

Protopia는 고객과 직접 협력하여 특정 사용 사례에 적합한 ALPHA 값을 결정합니다.

요소	세부
모델	FaceBoxes(PyTorch) -
데이터 세트	FDDB 데이터 세트

프로토피아 난독화	알파	정확성
아니요	해당 없음	0.9337148153739079
예	0.05	0.9028766627325002
예	0.1	0.9024301009661478
예	0.2	0.9081836283186224
예	0.4	0.9073066107482036
예	0.6	0.8847816568680239
예	0.8	0.8841195749171925
예	0.9	0.8455427675252052
예	0.95	0.8455427675252052

난독화 속도

이 검증을 위해 1920 x 1080 픽셀 이미지에 Protopia 난독화를 5번 적용하고 난독화 단계가 완료되는 데 걸리는 시간을 매번 측정했습니다.

단일 NVIDIA V100 GPU에서 실행되는 PyTorch를 사용하여 난독화를 적용하고, 실행 사이에 GPU 캐시를 지웠습니다. 난독화 단계는 5번의 실행에 걸쳐 각각 5.47ms, 5.27ms, 4.54ms, 5.24ms, 4.84ms가 걸렸습니다. 평균 속도는 5.072ms였습니다.

결론

데이터는 저장 중, 전송 중, 컴퓨팅 중이라는 세 가지 상태로 존재합니다. 모든 AI 추론 서비스에서 중요한 부분은 전체 프로세스 동안 위협으로부터 데이터를 보호하는 것입니다. 추론 중에 데이터를 보호하는 것은 매우 중요합니다. 추론 과정에서 외부 고객과 추론 서비스를 제공하는 기업에 대한 개인 정보가 노출될 수 있기 때문입니다. Protopia AI는 오늘날 시장에서 기밀 AI 추론을 위한 눈에 띄지 않는 소프트웨어 전용 솔루션입니다. Protopia를 사용하면 AI는 현재 AI/ML 작업을 수행하는 데 필수적인 데이터 레코드의 변환된 정보만 공급받고 그 이상은 공급받지 않습니다. 이러한 확률적 변환은 마스킹의 한 형태가 아니며, 큐레이트된 노이즈를 사용하여 데이터의 표현을 수학적으로 변경하는 데 기반을 둡니다.

ONTAP 기능을 갖춘 NetApp 스토리지 시스템은 로컬 SSD 스토리지와 동일하거나 더 나은 성능을 제공하며, NetApp DataOps Toolkit과 결합하면 데이터 과학자, 데이터 엔지니어, AI/ML 개발자, 비즈니스 또는 기업 IT 의사 결정권자에게 다음과 같은 이점을 제공합니다.

- AI 시스템, 분석 및 기타 중요 비즈니스 시스템 간에 데이터를 손쉽게 공유할 수 있습니다. 이러한 데이터 공유를 통해 인프라 오버헤드가 줄어들고, 성능이 향상되며, 기업 전체의 데이터 관리가 간소화됩니다.

- 비용을 최소화하고 리소스 사용을 개선하기 위해 독립적으로 확장 가능한 컴퓨팅 및 스토리지.
- 즉각적이고 공간 효율적인 사용자 작업 공간, 통합 버전 제어, 자동화된 배포를 위한 통합 스냅샷 복사본과 복제를 사용하여 개발 및 배포 워크플로를 간소화합니다.
- 재해 복구, 비즈니스 연속성 및 규정 요구 사항을 위한 엔터프라이즈급 데이터 보호 및 데이터 거버넌스.
- 데이터 관리 작업을 간편하게 호출할 수 있습니다. Jupyter Notebook의 NetApp DataOps Toolkit에서 데이터 과학자 작업 공간의 스냅샷 사본을 빠르게 가져와 백업하고 추적할 수 있습니다.

NetApp 및 Protopia 솔루션은 엔터프라이즈급 AI 추론 배포에 이상적인 유연하고 확장 가능한 아키텍처를 제공합니다. 이를 통해 데이터 보호가 가능해지고 민감한 정보에 대한 개인 정보 보호가 가능해져, 온프레미스와 하이브리드 클라우드 배포 모두에서 책임감 있는 AI 관행을 통해 기밀 AI 추론 요구 사항을 충족할 수 있습니다.

추가 정보 및 감사의 말씀을 찾을 수 있는 곳

이 문서에 설명된 정보에 대해 자세히 알아보려면 다음 문서 및/또는 웹사이트를 참조하세요.

- NetApp ONTAP 데이터 관리 소프트웨어 - ONTAP 정보 라이브러리

  http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286

- 컨테이너용 NetApp 영구 스토리지 NetApp Trident

  "https://netapp.io/persistent-storage-provisioner-for-kubernetes/"

- NetApp DataOps 툴킷

  "https://github.com/NetApp/netapp-dataops-toolkit"

- 컨테이너용 NetApp 영구 스토리지 NetApp Trident

  "https://netapp.io/persistent-storage-provisioner-for-kubernetes/"

- Protopia AI—기밀 추론

  "https://protopia.ai/blog/protopia-ai-takes-on-the-missing-link-in-ai-privacy-confidential-inference/"

- NetApp BlueXP 복사 및 동기화

  "https://docs.netapp.com/us-en/occm/concept_cloud_sync.html#how-cloud-sync-works"

- NVIDIA Triton 추론 서버

  "https://developer.nvidia.com/nvidia-triton-inference-server"

- NVIDIA Triton 추론 서버 설명서

  "https://docs.nvidia.com/deeplearning/triton-inference-server/index.html"

- PyTorch의 FaceBox

  "https://github.com/zisianw/FaceBoxes.PyTorch"

감사의 말

- Mark Cates, NetApp 수석 제품 관리자
- Sufian Ahmad, 기술 마케팅 엔지니어, NetApp
- Hadi Esmaeilzadeh, Protopia AI의 최고 기술 책임자 겸 교수

저작권 정보

Copyright © 2025 NetApp, Inc. All Rights Reserved. 미국에서 인쇄된 본 문서의 어떠한 부분도 저작권 소유자의 사전 서면 승인 없이는 어떠한 형식이나 수단(복사, 녹음, 녹화 또는 전자 검색 시스템에 저장하는 것을 비롯한 그래픽, 전자적 또는 기계적 방법)으로도 복제될 수 없습니다.

NetApp이 저작권을 가진 자료에 있는 소프트웨어에는 아래의 라이센스와 고지사항이 적용됩니다.

본 소프트웨어는 NetApp에 의해 '있는 그대로' 제공되며 상품성 및 특정 목적에의 적합성에 대한 명시적 또는 묵시적 보증을 포함하여(이에 제한되지 않음) 어떠한 보증도 하지 않습니다. NetApp은 대체품 또는 대체 서비스의 조달, 사용 불능, 데이터 손실, 이익 손실, 영업 중단을 포함하여(이에 국한되지 않음), 이 소프트웨어의 사용으로 인해 발생하는 모든 직접 및 간접 손해, 우발적 손해, 특별 손해, 징벌적 손해, 결과적 손해의 발생에 대하여 그 발생 이유, 책임론, 계약 여부, 엄격한 책임, 불법 행위(과실 또는 그렇지 않은 경우)와 관계없이 어떠한 책임도 지지 않으며, 이와 같은 손실의 발생 가능성이 통지되었다 하더라도 마찬가지입니다.

NetApp은 본 문서에 설명된 제품을 언제든지 예고 없이 변경할 권리를 보유합니다. NetApp은 NetApp의 명시적인 서면 동의를 받은 경우를 제외하고 본 문서에 설명된 제품을 사용하여 발생하는 어떠한 문제에도 책임을 지지 않습니다. 본 제품의 사용 또는 구매의 경우 NetApp에서는 어떠한 특허권, 상표권 또는 기타 지적 재산권이 적용되는 라이센스도 제공하지 않습니다.

본 설명서에 설명된 제품은 하나 이상의 미국 특허, 해외 특허 또는 출원 중인 특허로 보호됩니다.

제한적 권리 표시: 정부에 의한 사용, 복제 또는 공개에는 DFARS 252.227-7013(2014년 2월) 및 FAR 52.227-19(2007년 12월)의 기술 데이터-비상업적 품목에 대한 권리(Rights in Technical Data -Noncommercial Items) 조항의 하위 조항 (b)(3)에 설명된 제한사항이 적용됩니다.

여기에 포함된 데이터는 상업용 제품 및/또는 상업용 서비스(FAR 2.101에 정의)에 해당하며 NetApp, Inc.의 독점 자산입니다. 본 계약에 따라 제공되는 모든 NetApp 기술 데이터 및 컴퓨터 소프트웨어는 본질적으로 상업용이며 개인 비용만으로 개발되었습니다. 미국 정부는 데이터가 제공된 미국 계약과 관련하여 해당 계약을 지원하는 데에만 데이터에 대한 전 세계적으로 비독점적이고 양도할 수 없으며 재사용이 불가능하며 취소 불가능한 라이센스를 제한적으로 가집니다. 여기에 제공된 경우를 제외하고 NetApp, Inc.의 사전 서면 승인 없이는 이 데이터를 사용, 공개, 재생산, 수정, 수행 또는 표시할 수 없습니다. 미국 국방부에 대한 정부 라이센스는 DFARS 조항 252.227-7015(b)(2014년 2월)에 명시된 권한으로 제한됩니다.

상표 정보

NETAPP, NETAPP 로고 및 http://www.netapp.com/TM에 나열된 마크는 NetApp, Inc.의 상표입니다. 기타 회사 및 제품 이름은 해당 소유자의 상표일 수 있습니다.