Journal of KIIT. Vol. 24, No. 3, pp. 1-10, Mar. 31, 2026. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2026.24.3.1

딥페이크 탐지 학습의 효율성 향상을 위한 TFRecord 기반 데이터 파이프라인 기법

최승호*[1], 이석훈*[2]

An Efficient TFRecord-based Data Pipeline for Deepfake Detection Training

Seungho Choi*[1], Sukhoon Lee*[2]

이 연구는 과학기술정보통신부의 재원으로 한국지능정보사회진흥원의 지원을 받아 구축된 “딥페이크 변조영상”을 활용하여 수행된 연구입니다. 본 연구에 활용된 데이터는 AI허브(aihub.or.kr)에서 다운로드 받을 수 있습니다.

요 약

딥페이크 기술은 허위 정보 확산과 보안 위협을 초래하며, 이에 대한 탐지 기술의 중요성이 커지고 있다. 기존 연구가 정확도 향상에 집중한 반면, 데이터셋 규모와 파이프라인 구성에 따른 성능, 효율성의 균형은 충분히 논의되지 않았다. 본 연구는 Baseline, DiskCache, TFRecord 세 가지 파이프라인을 50만, 25만, 5만 장 데이터셋에서 비교하고, MobileNetV2 모델을 동일한 환경에서 학습하여 성능 지표와 학습 효율성을 분석하였다. 실험 결과, TFRecord는 정확도와 속도, 안정성 측면에서 균형을 이루며 가장 실용적인 파이프라인으로 확인되었다. 본 연구는 데이터 파이프라인 설계의 중요성을 강조하며, 향후 전처리 최적화, 도메인 일반화, 시스템 확장 연구를 제안한다.

Abstract

Deepfake technology poses increasing risks to misinformation, privacy, and security. While prior studies emphasized detection accuracy, the balance between performance and efficiency across dataset sizes and pipeline configurations has been less explored. This study compares three data loading strategies—Baseline, DiskCache, and TFRecord—on datasets of 500,000, 250,000, and 50,000 images using the MobileNetV2 model under consistent conditions. Experimental results show that TFRecord provides the most practical balance of accuracy, speed, and stability. These findings underscore the importance of pipeline design and point to future work in preprocessing optimization, domain generalization, and system scalability.

Keywords

deepfake detection, data pipeline optimization, tfrecord, diskcache

* 국립군산대학교 소프트웨어융합공학과(*[2] 교신저자)
- ORCID[1]: https://orcid.org/0009-0001-1299-5020
- ORCID[2]: https://orcid.org/0000-0002-3390-5602

· Received: Oct. 29, 2025, Revised: Mar. 12, 2026, Accepted: Mar. 15, 2026
· Corresponding Author: Sukhoon Lee
Dept. of Software Science & Engineering, Kunsan National University, Korea
Tel.: +82-63-469-8914, Email: leha82@kunsan.ac.kr

Ⅰ. 서 론

본 연구는 딥페이크 탐지 학습 환경에서 데이터 파이프라인 방식이 성능과 자원 효율성에 미치는 영향을 체계적으로 비교분석하는 것을 목적으로 한다. 기존 연구들이 주로 탐지 정확도 향상에 집중해 온 반면, 실제 적용 환경에서 중요한 학습 시간, 학습 효율성과 같은 지표는 상대적으로 간과되어 왔다. 따라서 본 연구는 성능과 효율성 간의 균형을 규명하고, 다양한 데이터셋 규모에서 가장 실용적인 파이프라인 방식을 제시하고자 한다.

딥페이크(Deepfake)는 최근 몇 년간 급격히 발전하며 사회 전반에 걸쳐 심각한 위협 요인으로 작용하고 있다[1][2]. 합성 영상은 기존 편집 기술로는 구현하기 어려운 사실감을 제공하며, 이는 허위 정보 유포, 정치적 선전, 사생활 침해, 금융 사기 등 다양한 문제를 야기한다[3]. 특히 고품질 합성 데이터와 공개된 학습 코드의 확산은 탐지 난이도를 크게 높이고 있어, 신뢰할 수 있는 탐지 기술의 필요성이 더욱 강조되고 있다.

이에 대응하여 다양한 탐지 기법이 제안되었다. 포렌식 신호 기반 분석[4], 멀티모달 구조[5], 캡슐 네트워크[6], 이중 스트림 신경망[7] 등은 성능 향상에 기여했으나, 대부분 정확도, 재현율과 같은 지표 개선에 집중하였다. 그러나 실제 연구 환경과 산업 현장에서는 GPU(Graphics Processing Unit), CPU (Central Processing Unit), RAM(Random Access Memory)과 같은 자원이 제한적이며, 특히 대규모 학습에서는 데이터 파이프라인의 구조와 효율성이 모델 성능 못지않게 중요한 요소로 작용한다[8][9].

대규모 데이터셋은 일반화 성능을 높일 수 있지만, 동시에 I/O 병목, GPU 유휴 시간, RAM 과부하 문제를 발생시킨다. 반대로 소규모 데이터셋은 학습 속도는 빠르지만, 성능 변동성과 일반화 저하 문제가 뒤따른다. 또한 데이터 로딩 방식 역시 중요한 변수로, Baseline 방식은 단순하지만 대규모 학습 시 디스크 병목을 일으킬 수 있고, DiskCache는 안정성을 제공하지만 메모리 사용량과 처리 시간이 크다[10][11]. 이에 비해 TFRecord는 시리얼 포맷 기반으로 효율적인 데이터 접근을 제공하며, 속도와 안정성의 균형을 유지할 수 있어 다양한 규모의 학습 환경에서 주목할 만하다.

본 연구는 딥페이크 탐지 시 학습의 효율성 향상을 위하여 TFRecord 기반 데이터 파이프라인 기법을 제안한다. 이를 위하여 데이터셋을 50만, 25만, 5만 장으로 구분하고 Baseline, DiskCache, TFRecord 세 가지 방식을 적용하여 성능과 효율성 간의 관계를 체계적으로 분석함으로써 실용적인 파이프라인 설계 방향을 제시한다. 각 조합에 대해 정확도, 정밀도, 재현율, F1-score와 같은 성능 지표뿐 아니라 학습 시간, 학습 효율성을 평가하였다. 이를 통해 성능과 효율성 간의 관계를 분석하고, 다양한 환경에서 실용적인 파이프라인 설계 방향을 제시하는 것을 목표로 한다.

본 논문의 구성은 다음과 같다. 제2장에서는 딥페이크 탐지 연구 동향과 데이터 파이프라인 관련 선행 연구를 정리한다. 제3장에서는 연구 방법과 실험 설계를 설명하며, 제4장에서는 데이터셋 규모와 파이프라인 방식에 따른 실험 결과를 제시한다. 마지막으로 제5장에서는 연구의 주요 발견을 종합하고, TFRecord 기반 파이프라인의 강점을 중심으로 향후 연구 방향을 제안한다.

Ⅱ. 관련 연구

2.1 딥페이크 탐지

딥페이크 탐지 연구는 공개 벤치마크 데이터셋의 등장과 함께 빠르게 발전하고 있다. 대표적으로 FaceForensics++는 다양한 합성 기법과 두 단계의 압축 수준을 포함한 방대한 영상 집합을 제공하여, 탐지 모델의 보편적 성능을 검증할 수 있는 표준으로 자리 잡고 있다[2]. 이어서 Celeb-DF는 실제 촬영 영상과 거의 구분하기 어려운 고품질 합성물을 제공함으로써 탐지 난이도를 크게 높였으며, 탐지기의 일반화 성능을 평가하는 데 중요한 역할을 하고 있다[3]. 또한 오디오, 비디오를 아우르는 멀티모달 벤치마크는 다양한 합성 조건과 품질 저하 요인을 반영하여 실제 환경을 모사하는데 기여하였다[5].

탐지 방법론 측면에서는 다양한 접근이 제안되었

다. Face X-Ray는 합성 방식이나 생성 기법에 무관하게 공통으로 나타나는 블렌딩 단서를 활용하여 범용 탐지를 가능하게 한다[4]. Capsule-forensics는 캡슐 네트워크의 구조적 특성을 이용해 합성 영역을 효과적으로 탐지하며, 기존 CNN(Convolutional Neural Network) 기반 탐지기의 한계를 보완한다[6]. 또한 Two-Stream 구조는 CNN 기반 고수준 특징과 스테가 분석 기반 저수준 통계 특징을 결합하여, 고품질, 고압축 데이터셋에서도 높은 강건성을 보여준다[7].

한편, 도메인 차이에 따른 일반화 문제를 해결하기 위한 연구가 활발히 진행되고 있다. 선택적 도메인 불변 특징 학습 기법은 스타일과 콘텐츠를 분리하여 불필요한 도메인 특성을 배제함으로써 다양한 합성 방식을 아우르는 일반성을 확보하였다[8]. 한편, CNN과 RNN(Recurrent Neural Network)을 혼합한 시공간 구조는 영상 내 시공간적 단서를 동시에 학습하여 탐지 성능을 향상 시키며[5], 오디오, 비디오 동기성 검증 기법은 멀티모달 접근을 통해 실제 환경에서의 일반화 성능 개선에 기여하였다[12].

효율성과 실시간성을 고려한 연구도 주목받고 있다. 이진 신경망(BNN, Binary Neural Network) 기반 모델은 연산량과 메모리 소모를 크게 줄이면서도 일정 수준의 정확도를 유지할 수 있으며[9], 후처리 기법과 결합하면 실시간 응용에도 활용 가능하다[13]. 또한 딥러닝 기반 탐지기가 적대적 공격에 취약하므로, 적대적 학습이나 랜덤화 기법을 통한 강건성 보완 연구도 병행되고 있다[14].

그러나 이러한 연구들은 딥페이크 탐지를 위하여 단순히 정확도 개선에 뿐 아니라, 데이터 다양성, 도메인 일반화, 효율성, 강건성까지 복합적으로 고려해야 함을 보여준다.

2.2 데이터 파이프라인과 데이터 로딩 방식

딥러닝 학습 성능은 모델 아키텍처뿐 아니라 데이터 파이프라인의 구성에 크게 의존한다. 특히 대규모 학습 환경에서는 I/O 병목, GPU 활용률 저하, RAM 과다 점유와 같은 문제가 빈번하게 발생하며, 이를 완화하기 위한 다양한 연구가 진행되어 왔다.

예를 들어, 분산 학습 프레임워크 성능 모델링 연구에서는 입력 파이프라인의 튜닝과 리소스 스케줄링이 GPU 유휴 시간 감소의 핵심임을 보여주었고[11], 계산, 저장 자원 분리 환경에서는 캐시 대역폭 최적화가 학습 비용 절감과 효율성 향상에 중요한 요소임이 보고되었다[10].

데이터 로딩 방식은 이러한 병목 현상을 완화하기 위한 핵심 전략으로, 파이프라인 구성에 따라 성능과 자원 사용량이 크게 달라진다. Baseline 방식은 원본 이미지를 학습 중에 직접 디스크에서 불러오는 구조로, 구현이 단순하고 직관적이지만 대규모 학습에서는 디스크 접근이 병목으로 작용할 수 있다[11].

TFRecord 방식은 TensorFlow에서 제공하는 시리얼라이즈된 바이너리 포맷으로, 데이터를 하나의 파일 또는 여러 개의 샤드로 저장한 후 스트리밍 방식으로 읽어 들인다. 이러한 구조는 디스크 접근 횟수를 줄이고, prefetch, 병렬 로딩(Parallel loading), 셔플(Shuffle) 등의 고급 파이프라인 연산과 결합되어 GPU와 CPU 간의 처리 병목을 완화할 수 있다. 특히 셔플 버퍼 크기와 병렬 읽기 설정은 학습 성능에 직접적인 영향을 미치며, 적절한 튜닝을 통해 학습 속도와 정확도를 동시에 개선할 수 있다[15].

DiskCache 방식은 이미지 데이터를 메모리 기반 캐시 계층에 저장한 후 접근하는 구조로 제안되었으며[11], 디스크 접근 병목을 줄일 수 있다는 장점이 있다. 다만 메모리 사용량 증가나 속도 저하와 같은 한계는 본 연구의 실험 결과에서 다시 논의한다.

따라서 데이터 파이프라인은 단순한 입력 경로가 아니라 학습 시간, 학습 효율성과 직결되는 핵심 변수이다. 최근 연구에서는 하드웨어 기반 최적화나 파이프라인 병렬화 기법을 통해 전체 시스템 처리량을 개선하는 방법도 논의되고 있으며[16], 본 연구 역시 이러한 맥락에서 Baseline, TFRecord, DiskCache를 비교하여 실제 딥페이크 탐지 학습 환경에서 성능과 자원 효율성에 미치는 영향을 실험적으로 검증한다.

III. 제안 기법

3.1 전체 파이프라인 개요

본 연구에서 제안하는 딥페이크 탐지 학습 파이프라인은 그림 1과 같은 순서로 구성된다.

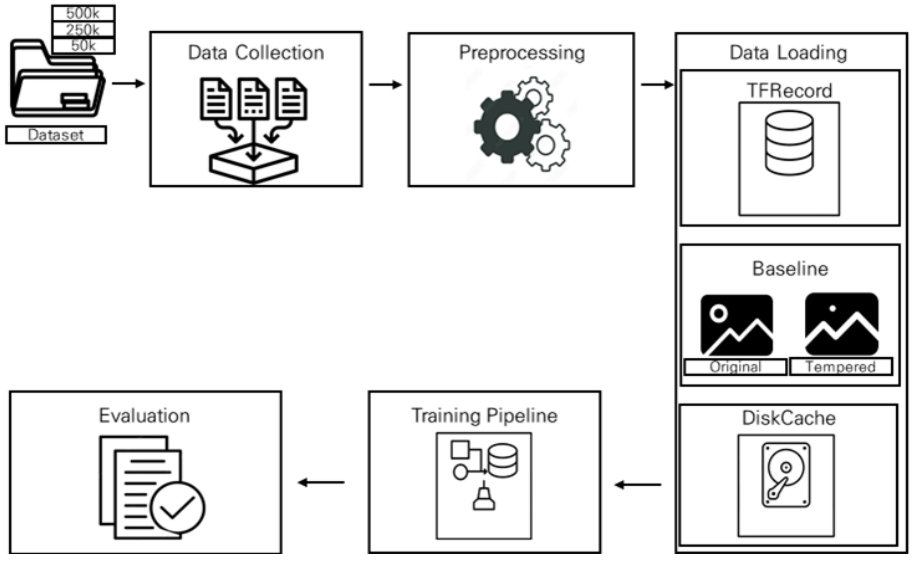

그림 1. 데이터 파이프라인 학습 절차
Fig. 1. DataPipeline process

먼저 원천 영상 데이터로부터 1초 단위의 이미지 샘플을 추출하고, 이를 전처리하여 학습에 적합한 형태로 정제한 후 TFRecord 포맷으로 변환한다. 이 과정에서 전체 데이터셋 50만 장 기준으로 25만, 5만 장으로 점진적으로 샘플링하여 학습에 투입함으로써, 데이트 크기 변화가 탐지 성능 및 학습 시간에 미치는 영향을 실험적으로 분석하였다.

본 연구에서는 학습 속도와 처리 안정성의 균형을 평가하기 위해 TFRecord를 핵심 파이프라인으로 제안하였으며, 이에 대한 객관적인 성능 비교를 위해 Baseline과 DiskCache를 대조군으로 설정하였다.

3.2 데이터 로딩 기법

데이터 로딩 기법은 딥페이크 탐지 학습 환경에서 I/O 병목 해소와 학습 안정성 확보를 위해 세 가지 방식으로 구성하였다. 우선 AIHub “딥페이크 변조영상” 데이터셋을 기반으로 원본 동영상에서 1초 간격으로 5프레임을 추출하여 JPEG로 저장하였다. 추출된 프레임은 변조 여부에 따라 Original과 Tempered 클래스에 배정되며, 학습, 검증, 시험 데이터를 8:1:1 비율로 분할하고 동일 영상 프레임이 중복되지 않도록 하여 데이터 누수를 방지하였다.

모든 이미지는 RGB 채널로 변환 후 224x224 해상도로 리사이즈하고, 픽셀 값을 [0,1] 범위로 정규화하였다.

3.2.1 Baseline 방식

Baseline 방식은 파일 시스템에서 원본 이미지를 학습 중에 직접 읽어 들이는 가장 단순한 구조이다. 구현이 용이하고 추가 저장 공간이 필요 없으며, 실험 초기 단계나 소규모 데이터셋에서는 빠르게 적용할 수 있는 장점이 있다. 그러나 이미지 파일을 매 반복마다 디스크에서 직접 불러오기 때문에 디스크 I/O 병목이 발생하기 쉽고, 특히 대규모 학습 환경에서는 GPU가 데이터를 기다리는 유휴 시간이 증가하여 전체 처리량이 저하된다. 또한 디스크 접근 순서나 파일 시스템의 상태에 따라 성능 편차가 발생할 수 있어 재현성 측면에서도 불리하다. 이러한 구조는 단순성과 접근성은 높지만, 확장성과 안정성 측면에서 한계가 있다[11].

3.2.2 DiskCache 방식

DiskCache 방식은 처음 읽은 이미지를 메모리 기

반 캐시 계층에 유지하여 디스크 I/O 병목을 완화하는 구조이다. 학습 초기에 디스크에서 이미지를 불러온 후, 이후 반복에서는 메모리에서 직접 접근함으로써 처리 속도의 안정성을 확보할 수 있다.

이 방식은 특히 대규모 데이터셋에서 낮은 분산과 높은 재현성을 보이며, 실험 반복 간 성능 편차가 적다는 장점이 있다. 그러나 전체 데이터를 메모리에 적재해야 하므로 RAM 사용량이 급증하며, 시스템 메모리 한계를 초과할 경우 캐시 관리 오버헤드가 발생하고 처리 속도가 오히려 저하될 수 있다. 또한 캐시 초기화나 관리 방식에 따라 성능이 달라질 수 있어, 실험 환경에 따라 세심한 조정이 필요하다[17].

3.2.3 TFRecord 방식

TFRecord 방식은 TensorFlow에서 제공하는 시리얼라이즈된 바이너리 포맷으로 스트리밍 방식으로 읽어들인다. 이 구조는 디스크 접근 횟수를 최소화하고, shuffle, prefetch, interleave 같은 파이프라인 연산과 병렬 읽기를 결합하여 GPU와 CPU 간 병목을 완화할 수 있다. 특히 셔플 버퍼 크기와 병렬 읽기 스레드 수를 튜닝하면 학습 속도와 분류 정확도를 동시에 개선할 수 있으며, 대규모 데이터셋에서도 안정적인 처리 성능을 유지할 수 있다. TFRecord는 저장 용량 측면에서도 서브선형 증가 경향을 보이며, 데이터 수가 증가해도 저장 공간 부담이 상대적으로 낮아 확장성 측면에서 유리하다. TFRecord는 대규모 데이터 처리에 최적화된 포맷으로 널리 알려져 있으나[9], 본 연구는 딥페이크 탐지라는 자원 집약적 도메인에 적용하여, 데이터셋 규모 변화에 따른 분류 성능과 학습 효율성 간의 실증적 상관관계를 규명했다는 점에서 차별화된 학술적 의의를 갖는다.

3.3 학습 모델

본 연구에서는 그림 2와 같이 MobileNetV2를 학습 모델로 채택하였다. MobileNetV2는 경량화된 합성곱 신경망 구조로, Depthwise Separable Convolution과 Inverted Residual Block을 기반으로 설계되어 연산량과 파라미터 수를 크게 줄이면서도 높은 분류 정확도를 유지할 수 있다. 이러한 구조적 특성은 메모리 사용량과 처리 속도를 동시에 고려해야 하는 반복 실험 환경이나 실제 배포 시나리오에서 특히 유리하다[9].

또한 MobileNetV2는 ResNet, VGG 등 고용량 네트워크에 비해 학습 시간이 짧고 GPU 자원 소모가 적으며, 상대적으로 작은 데이터셋에서도 안정적인 성능을 보이는 것으로 알려져 있다[18]. 다양한 데이터 파이프라인 구성에 대한 민감도가 낮아, 데이터 로딩 전략에 따른 성능 편차를 비교하는 본 연구의 목적에도 적합하다.

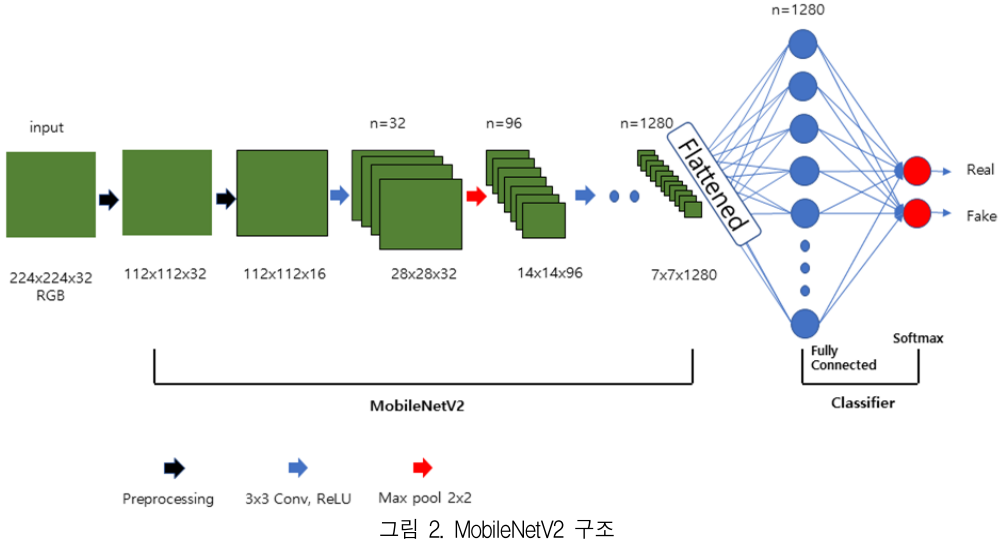

그림 2. MobileNetV2 구조
Fig. 2. MobileNetV2 structure

Ⅳ. 실험 및 평가

4.1 실험 방법

실험 설정은 다음과 같다. 사전학습 가중치는 사용하지 않았으며, 옵티마이저는 Adam, 초기 학습률은 0.001으로 설정하였다. 손실 함수는 Binary-Cross-Entropy를 적용하였고, 배치 크기는 32, 학습 epoch 수는 10으로 설정하였다. 모든 실험은 재현성을 위해 동일 조건에서 5회 독립적으로 반복 실행하였다.

모든 파이프라인은 동일한 데이터 분할과 전처리를 거쳐 학습 데이터셋에 공급되었다. 각 실험은 모델과 시스템 환경을 초기화하여 독립적으로 수행함으로써 데이터 로딩 순서, 캐싱 상태, GPU 메모리 점유와 같은 환경 요인이 결과에 영향을 주지 않도록 하였다. 성능 평가는 학습 과정에서 검증 데이터셋으로 모니터링하고, 최종적으로 테스트 데이터셋에서 수행되었다.

실험은 CPU는 Intel i9-13900k, RAM은 32GB, GPU는 NVIDIA RTX A6000이 장착된 서버 환경에서 수행되었으며, 본 연구에서 사용한 하드웨어 환경은 학습 시간과 이미지 처리 속도 등 효율성 지표를 해석하는데 참고 기준을 제공한다.

본 연구에서는 딥페이크 탐지를 이진 분류 문제로 정의한다. 여기서 Positive는 딥페이크 영상, Negative는 정상(Real) 영상으로 설정한다. 이에 따라 혼동행렬의 요소를 표 1과 같이 정의하였다.

표 1. 혼동 행렬 구성 요소 정의
Table 1. Definition of confusion matrtix components

Classification	Definition
True Positive (TP)	Correctly detecting a deepfake video as a deepfake
True Negative (TN)	Correctly detecting a real video as real
False Positive (FP)	Incorrectly detecting a real video as a deepfake
False Negative (FN)	Incorrectly detecting a deepfake video as real

평가 지표는 크게 성능 지표와 효율성 지표의 두 가지 범주로 나누어 측정한다. 성능 지표는 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1-Score를 측정하며, 각 수식은 식 (1)~(4)와 같다.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\mathrm{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - score = \frac{2 \bullet Precision \bullet Recall}{\mathrm{Precision} + Recall} \quad (4)$$

효율성 지표는 학습 과정에서 소요되는 전체 학습 시간과 단위 시간당 처리 가능한 이미지 수(img/s)를 측정한다. 수식은 식 (5)와 같다.

$$Efficiency = \frac{Accuracy}{Training\ Time(\min)} \quad (5)$$

또한 모든 지표는 5회 독립 반복 실험을 통해 평균과 표준편차를 산출하여, 성능과 효율성뿐 아니라 안정성까지 평가하였다. 이를 통해 TFRecord 기반 파이프라인의 실질적 우수성을 종합적으로 검증하였다.

4.2 평가 결과

세 가지 데이터 파이프라인의 분류 성능을 데이터셋 규모별로 비교하였다. 표 2은 50만, 25만, 5만 장에 따른 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1-score 결과를, 그림 3은 동일한 결과 중 Accuracy를 시각적으로 비교한 그래프이다.

50만 장에서는 DiskCache가 Accuracy 0.9599, F1-score 0.9599로 가장 높은 성능을 기록하였다. TFRecord는 Accuracy 0.9108, F1-score 0.9100으로 DiskCache 보다는 낮았으나 Baseline 대비 뚜렷한 개선을 보였다.

25만 장에서는 Baseline이 Accuracy 0.8896, F1-score 0.8847로 가장 높은 성능을 보였다. 반면 DiskCache와 TFRecord는 각각 Accuracy 0.6933, 0.6980으로 Baseline 대비 큰 격차를 나타냈다.

5만 장에서는 Baseline이 Accuracy 0.7866, F1-score 0.7827로 가장 높은 평균 성능을 보였으며, TFRecord는 Accuracy 0.7651, F1-score 0.7600으로 근접하였다. 다만 Baseline은 반복 실험 간 변동폭이 큰 반면, TFRecord는 상대적으로 안정적인 성능을 유지하였다. DiskCache는 Accuracy 0.6839, F1-score 0.6491로 가장 낮은 성능을 보였다.

종합하면, 데이터셋 규모에 따라 파이프라인 성능이 달라지는 양상을 명확히 보여준다. 대규모에서는 DiskCache가 우세하나, 중소규모에서는 Baseline이 더 적합하며, TFRecord는 전반적으로 안정적인 성능을 유지하는 특징을 보였다.

표 2. 데이터셋 규모별 파이프라인 성능 비교
Table 2. Pipeline performance by dataset size

Size	Pipe.	Acc.	Prec.	Rec.	F1
500k	Base.	0.8506	0.8942	0.8506	0.8386
	DCache	0.9599	0.9622	0.9599	0.9599
	TFRec.	0.9108	0.9192	0.9108	0.9100
250k	Base.	0.8896	0.9132	0.8896	0.8847
	DCache	0.6933	0.7953	0.6933	0.6616
	TFRec.	0.6980	0.7897	0.6980	0.6626
50k	Base.	0.7866	0.8096	0.7866	0.7827
	DCache	0.6839	0.7537	0.6839	0.6491
	TFRec.	0.7651	0.7874	0.7651	0.7600

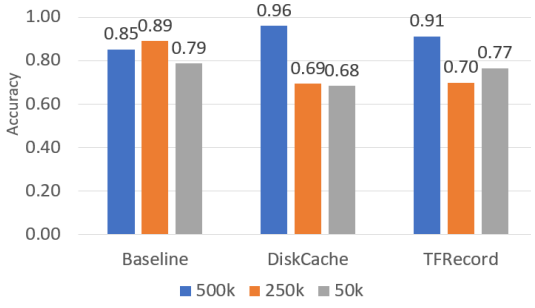

그림 3. 데이터셋 규모별 파이프라인 Accuracy 비교
Fig. 3. Accuracy comparison of data loading oipelines by dataset size

한편, 효율성에서는 성능과 다른 양상이 나타났다. 표 3는 데이터셋 크기 50만, 25만, 5만장에서 측정한 학습 시간과 Accuracy를 기반으로 산출한 효율성 지표를 정리한 결과이며, 그림 4는 동일한 결과 중 학습 시간을 시각적으로 비교한 그래프이다. 50만 장의 경우 DiskCache의 정확도가 가장 높았으나 학습 시간이 TFRecord 대비 약 2.5배 더 소요되어 효율성 지표는 약 42% 수준에 불과하였다. 25만장과 5만장에서도 Baseline과 TFRecord가 대등하게 높은 효율성을 보인 반면, DiskCache는 TFRecord 대비 65~68% 낮은 수치를 기록하며 크게 뒤처졌다.

표 3. 데이터셋 규모별 파이프라인 학습 효율성 비교
Table 3. Pipeline training efficiency by dataset size

Size	Pipe.	Time (min)	Acc.	Speed (img/sec)	Eff.
500 k	Base.	106.22	0.8506	628.0	0.00801
	DCache	269.53	0.9599	247.2	0.00356
	TFRec.	108.32	0.9108	615.8	0.00841
250 k	Base.	57.26	0.8896	582.3	0.01552
	DCache	153.68	0.6933	216.9	0.00451
	TFRec.	53.71	0.6980	619.3	0.01299
50k	Base.	11.09	0.7866	602.0	0.0709
	DCache	30.83	0.6839	216.2	0.0222
	TFRec.	10.87	0.7651	614.2	0.0704

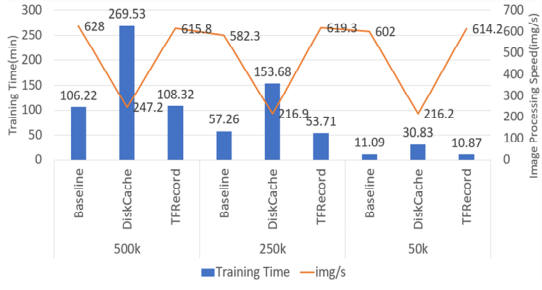

그림 4. 데이터셋 규모별 파이프라인 학습 시간 비교
Fig. 4. Training time comparison of data loading pipelines by dataset size

종합하면, 데이터셋 크기에 따라 파이프라인의 효율성이 크게 달라지는 양상을 보여준다. DiskCache는 정확도 측면에서는 우수했으나, 대규모 데이터 처리 시 메모리(RAM) 과부하를 유발하여 효율성 지표에서는 다른 방식보다 2.5~3배 낮은 결과를 보였다. Baseline 역시 매 스텝마다 디스크 I/O 병목을 발생시켜 대규모 학습에 취약한 한계를 나타냈다. 반면, TFRecord는 데이터를 바이너리 형태로 스트리밍하고 병렬 처리를 수행함으로써 CPU-GPU간 데이터 전송 대기 시간을 최소화하는 구조적 이점을 지닌다. 결과적으로 TFRecord는 데이터 규모 변화에도 하드웨어 병목에 빠지지 않고

성능과 학습 시간을 균형 있게 유지하며, Baseline보다 안정적이고 재현성이 높아 전반적으로 가장 실용적인 선택지로 확인되었다.

V. 결 론

본 연구에서는 딥페이크 탐지 모델 학습에서 데이터 파이프라인 방식이 성능과 효율성에 미치는 영향을 50만, 25만, 5만 장 데이터셋을 대상으로 정량적으로 비교분석하였다. 세 가지 전략 중 TFRecord 기반 파이프라인은 DiskCache보다 약간 낮은 정확도를 보였으나, 학습 속도와 처리량에서 우수했으며, Baseline보다 높은 안정성과 재현성을 동시에 달성하였다.

기존 연구들이 딥페이크 탐지의 정확도 향상에만 치중했던 것과 달리, 본 연구는 파이프라인 구성에 따라 탐지 시스템을 구축할 때 단순히 정확도만 고려한 것이 아니라, 처리속도, 저장 효율성, 재현성 등 다각도의 균형을 함께 평가해야함을 시사한다. TFRecord는 이러한 종합적 관점에서 가장 실용적인 선택지로, 연구실, 산업 현장 등 다양한 적용 시나리오에서 효율적이고 안정적인 학습 파이프라인을 제공할 것이다.

향후 연구에서는 전처리 단계의 압축, 해상도 변환 기법 최적화[19][20], 합성 기법별 도메인 일반화 전략[21], 하드웨어, 소프트웨어 공동 최적화를 통한 파이프라인 가속화, 오디오, 비디오 멀티모달 확장 등으로 범위를 넓혀, TFRecord기반 구조의 확장성과 강건성을 더욱 강화할 필요가 있다.

References

[1] L. Verdoliva, "Media forensics and deepfakes: An overview", IEEE Journal of Selected Topics in Signal Processing, Vol. 14, No. 5, pp. 1231-1244, Aug. 2020. https://doi.org/10.1109/JSTSP.2020.3002101.

[2] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies, and M. Nießner, "FaceForensics++: Learning to detect manipulated facial images", Proc. IEEE/CVF Int. Conf. Computer Vision (ICCV), Seoul, Korea, pp. 1-11, Oct. 2019. https://doi.org/10.1109/ICCV.2019.00009.

[3] Y. Li, P. Sun, H. Qi, and S. Lyu, "Celeb-DF: A large-scale challenging dataset for deepfake forensics", Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, pp. 3207-3216, Jun. 2020. https://doi.org/10.1109/CVPR42600.2020.00327.

[4] R. Lanzino, F. Fontana, A. Diko, M. R. Marini, and L. Cinque, "Faster Than Lies: Real-time Deepfake Detection using Binary Neural Networks", Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA,, pp. 3771-3780, Jun. 2024.

[5] J. Lomnitz, A. S. Martinez, and M. R. Gupta, "Multimodal approach for deepfake detection", Proc. IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Washington DC, DC, USA, pp. 1-6, Oct. 2020. https://doi.org/10.1109/AIPR50011.2020.9425192.

[6] H. H. Nguyen, F. Fang, J. Yamagishi, and I. Echizen, "Capsule-forensics: Using capsule networks to detect forged images and videos", Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, pp. 2927-2931, May 2019. https://doi.org/10.1109/ICASSP.2019.8682602.

[7] P. Zhou, X. Han, V. I. Morariu, and L. S. Davis, "Two-Stream Neural Networks for Tampered Face Detection", Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, pp. 1831-1839, Jul. 2017. https://doi.org/10.1109/CVPRW.2017.229.

[8] C. Lai, Y. Li, and S. Lyu, "Selective domain-invariant feature for generalizable deepfake detection", Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP), Seoul, Korea, pp. 1-5, Apr. 2024. https://doi.org/10.1109/ICASSP48485.2024.10447889.

[9] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, et al., "TensorFlow: A system for large-scale machine learning", Proc. USENIX

Symp. Operating Systems Design and Implementation (OSDI), Savannah, GA, USA, pp. 265-283, Nov. 2016.

[10] Y. Li, J. Zhang, and K. Chen, "Dynamic resource allocation for deep learning clusters with separated compute and storage", Proc. IEEE Conf. Computer Communications (INFOCOM), New York City, NY, USA, pp. 1-9, May 2023. https://doi.org/10.1109/INFOCOM53939.2023.10228920.

[11] S. Shi, Q. Wang, and X. Chu, "Performance modeling and evaluation of distributed deep learning frameworks on GPUs", IEEE Transactions on Parallel and Distributed Systems, Athens, Greece, Vol. 29, No. 3, pp. 698-709, Mar. 2018. https://doi.org/10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.000-4.

[12] P. A. Battula, R. Kumar, and S. S. Reddy, "Deepfake detection using convolutional neural networks and recurrent neural networks", Proc. Int. Conf. Distributed Computing and Intelligent Computing (ICDICI), pp. 1-6, Nov. 2024. https://doi.org/10.1109/ICDICI62993.2024.10810970

[13] G. Akyol, M. E. Karsligil, and A. E. Cetin, "Deep learning-based real-time object detection for autonomous driving", Proc. IEEE Applied Imagery Pattern Recognition Workshop (AIPR), Tirunelveli, India, pp. 1-6, Oct. 2020. https://doi.org/10.1109/SIU49456.2020.9302500.

[14] A. Devasthale and S. Sural, "Adversarially robust deepfake video detection", Proc. IEEE Symp. Series on Computational Intelligence (SSCI), Singapore, Singapore, pp. 1-8, Dec. 2022. https://doi.org/10.1109/SSCI51031.2022.10022079.

[15] TensorFlow TFRecord tutorial, https://www.tensorflow.org/tutorials/load_data/tfrecord?hl=ko. [accessed: Oct. 17, 2025]

[16] Y. Duan and J. Wu, "Optimizing Resource Allocation in Pipeline Parallelism for Distributed DNN Training", Proc. IEEE Int. Conf. Parallel and Distributed Systems (ICPADS), Nanjing, China, pp. 1-8, Jan. 2023. https://doi.org/10.1109/ICPADS56603.2022.00029.

[17] J. Zhou, F. Chen, Q. He, X. Xia, R. Wang, and Y. Xiang, "Data Caching Optimization With Fairness in Mobile Edge Computing", IEEE Transactions on Services Computing, Vol. 16, No. 3, pp. 1750-1762, May 2023. https://doi.org/10.1109/TSC.2022.3197881.

[18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks", Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, pp. 4510-4520, Jun. 2018. https://doi.org/10.1109/CVPR.2018.00474.

[19] E. N. Wanyonyi and N. W. Masinde, "The Impact of Data Preprocessing on Machine Learning Model Performance: A Comprehensive Examination", International Journal of Scientific Research in Computer Science, Engineering and Information Technology, Vol. 11, No. 2, pp. 3814-3827, Apr. 2025. https://doi.org/10.32628/CSEIT25112854.

[20] C. Guerra-García, H. G. Pérez-González, F. Martínez-Pérez, R. Juárez-Ramírez, and S. Jiménez, "Applying Mechanisms of Data Profiling for Assuring Data Quality in the Software: A First Approach", Proc. 11th Int. Conf. Software Engineering Research and Innovation (CONISOFT), León, Guanajuato, Mexico, pp. 108-115, Nov. 2023. https://doi.org/10.1109/CONISOFT58849.2023.00023.

[21] T. T. Nguyen, Q. V. H. Nguyen, D. T. Nguyen, D. T. Nguyen, T. Huynh-The, S. Nahavandi, T. T. Nguyen, Q.-V. Pham, and C. M. Nguyen, "Deep learning for deepfakes creation and detection: A survey", Computer Vision and Image Understanding, Vol. 223, pp. 103525, Oct. 2022. https://doi.org/10.1016/j.cviu.2022.103525.

[22] L. Liu, Y. Wu, W. Wei, W. Cao, S. Sahin, and Q. Zhang, "Benchmarking deep learning frameworks: Design considerations, metrics and beyond", Proc. IEEE 38th Int. Conf. Distributed

Computing Systems (ICDCS), Vienna, Austria, pp. 1258-1269, Jul. 2018. https://doi.org/10.1109/ICDCS.2018.00125.

[23] L. Li, J. Bao, T. Zhang, H. Yang, D. Chen, F. Wen, and B. Guo, "Face X-ray for more general face forgery detection", Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), Online, pp. 5001-5010, Jun. 2020.

[24] S. Agarwal and H. Farid, "Photo forensics from JPEG dimples", Proc. IEEE Workshop on Information Forensics and Security (WIFS), Rennes, France, pp. 1-6, Dec. 2017. https://doi.org/10.1109/WIFS.2017.8267641.

[25] M. Ali, S. Roy, U. Saxena, T. Sharma, A. Raghunathan, and K. Roy, "Compute-in-Memory Technologies and Architectures for Deep Learning Workloads", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 30, No. 11, pp. 1615-1630, Nov. 2022. https://doi.org/10.1109/TVLSI.2022.3203583.

[26] H. Kim, J. Lee, and S. Park, "FLSNet: Robust facial landmark semantic segmentation", IEEE Access, Vol. 8, pp. 1-10, Jun. 2020. https://doi.org/10.1109/ACCESS.2020.3004359

[27] M. Gupta and P. Chandra, "A comparative study of clustering algorithms", Proc. IEEE Int. Conf. Computing, Communication and Automation (ICCCA), New Delhi, India, pp. 1-6, Mar. 2019.

저자소개

최 승 호 (Seungho Choi)

2013년 2월 : 가톨릭관동대학교 경영학과(경영학사)
2017년 2월 : 가톨릭관동대학교 경영학과(경영석사)
2024년 2월 ~ 현재 : 국립군산대학교 소프트웨어융합공학과 박사과정
관심분야 : 소프트웨어 공학, 딥페이크 탐지, 빅데이터 분석

이 석 훈 (Sukhoon Lee)

2009년 2월 : 고려대학교 전자및정보공학부(학사)
2011년 2월 : 고려대학교 컴퓨터·전파통신공학과(공학석사)
2016년 2월 : 고려대학교 컴퓨터·전파통신공학과(공학박사)
2016년 3월 ~ 2017년 3월 : 아주대학교 의료정보학과 연구강사
2017년 4월 ~ 현재 : 국립군산대학교 소프트웨어학과 부교수
관심분야 : 사물인터넷, 메타데이터 레지스트리, 데이터 품질, 연합 학습