

LightKD-Detector: An Empirical Investigation into Lightweight Deepfake Detection via Knowledge Distillation under Computational Constraints

MSc Research Project
MSc. Cyber Security

Govind Nhaloor
Student ID: 24104973

School of Computing
National College of Ireland

Supervisor: Mr. Liam McCabe

National College of Ireland
MSc Project Submission Sheet
School of Computing



Student Name: Govind Nhaloor
Student ID: 24104973
Programme: MSc. Cyber Security **Year:** 2025
Module: MSc (Research) Practicum
Supervisor: Mr. Liam McCabe
Submission Due Date: 10-12-25
Project Title: LightKD-Detector: An Empirical Investigation into Lightweight Deepfake Detection via Knowledge Distillation under Computational Constraints
Word Count: **10017** **Page Count:** **23**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:

A handwritten signature in blue ink, appearing to read "Govind Nhaloor".

Date: 10-12-25

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Table of Contents

No.	Title	Pg.no
	Abstract	2
1	Introduction	2
1.1	Background and Context	3
1.2	Motivation and Problem Statement	3
1.3	Research question and objectives	3
1.4	Contribution	3
1.5	Limitations and assumptions	3
1.6	Structure of report	4
2	Literature review	4
2.1	The socio technical crisis of synthetic media	4
2.2	The evolution of deep fake generation technologies	5
2.3	Deep learning-based detection architectures	6
2.4	The human algorithm interface and legal frameworks	7
2.5	The resource gap and adversarial robustness	8
3	Research methodology	9
3.1	Data gathering	9
3.2	Preprocessing and transformation	10
3.3	Implementation and solution development	10
3.4	Evaluation criteria	11
4	Design specification	11
4.1	High level system architecture	12
4.2	Detailed component design	12
4.3	Hardware and software stack	14
4.4	Evaluation design	14
5	Implementation	14
5.1	Development environment	14
5.2	Data pipeline implementation	15
5.3	Model training implementation	15
5.4	Interface and deployment implementation	16
5.5	Challenges encountered	17
6	Results and discussions	17
6.1	Computational efficiency analysis	17
6.2	Detection performance analysis	19
6.3	Qualitative visual analysis	19
7	Conclusion and future works	20
8	References	20

Table of Tables

No.	Title	Pg.no
1	Performance of Considered Models	17
2	Evaluation metrics of the final Student Model	18

Table of Figures

No.	Title	Pg.no
1	The Knowledge Distillation Engine	13
2	Custom Dataset Loader	15
3	Parameters considered for each model	17
4	Size of Each Model	17
5	Inference Time of Each Model	18
6	Throughput of Each Model	18
7	Detection of the student model.	19

LightKD-Detector: An Empirical Investigation into Lightweight Deepfake Detection via Knowledge Distillation under Computational Constraints

Govind Nhaloor¹

¹ National College of Ireland, Mayor Street, IFSC, Dublin 1, Ireland.

Abstract.

The introduction of Generative Adversarial Networks (GANs) in 2014, and the consequent rise of Diffusion Models in 2020, hyper-realism of deepfakes has spread, becoming a threat to the reliability and wholesomeness of information in society. Despite hardware improvements between the years 2018 and 2025, deepfake generation continues to outperform the detection capabilities on consumer devices. As of 2025, the major model for detection of deepfakes are run on server grade hardware with strong computing power and are not deployable on consumer level hardware where misinformation is spread rapidly. There is an immediate need to eradicate the trade-off between the size of the model, detection accuracy and the speed of inference. The proposed research relies on LightKD-Detector, a model that utilizes Knowledge Distillation (KD) to load knowledge of a heavy model (EfficientNet-B4) onto a lightweight model (MobileNetV2). The models had been trained on the FaceForensics++ dataset and then evaluated on the more challenging Celeb-DF (v2) dataset at a consumer grade GPU (RTX 3060) to replicate potential resource constraints in the real world. The intended efficiency was exceeded with the proposed MobileNetV2 Student model, achieving an impressive inference rate and a light model. However, the model collapsed in terms of mode collapse in its detections. These results are crucial indicators of the "Resource Gap" of AI forensics. As shown in the paper, though lightweight architectures resolve the issue of latency, the success of Knowledge Distillation is inevitably tied to the convergence of the Teacher model that is not possible in resource-constrained systems without pre-trained weights. This experimentally validates the fact that the entry barrier to strong deepfake defense is still low because of its complexity.

Keywords - Deepfake Detection, Knowledge Distillation, MobileNetV2, AI Forensics, Resource Constraints, Generative AI.

1. Introduction

1.1 Background and Context

The art of manipulating photographs has been a practice since the end of 20th century. The emergence of generative artificial intelligence (AI) has made the generation of synthetic media, also known as deepfakes where someone has substituted or distorted a person's image with high accuracy, accessible to everyone. According to Mohammed (2024), deepfake technology started attracting the attention of the world audience in the year 2017, as autoencoder-based algorithms were introduced to online forums. The field has since developed at a very high rate. As of 2025, with the release of Generative Adversarial Networks (GANs), followed by the subsequent development of Diffusion Models (DMs), in 2014 and early 2020s, respectively, deepfakes have not only become not just a technical curiosity anymore, but a very powerful weapon with the potential of creating hyper-realistic audio, video and text (Zhang et al., 2025).

These technologies are being used in the entertainment industry and in educational institutions with justifiable purposes, but there is also a high degree of danger in their dual use. As Gilbert and Gilbert (2024) point out, deepfakes are becoming more weaponized. They are being used to steal identities, commit financial fraud, and spread disinformation. This is not hypothetical weaponization, according to the recent surveys, the level of trust in the digital media is fading among the population. This is an essential crisis in the consumption of information, as Mohamed and Ozad (2025) discovered that 98% of respondents in wealthy groups now question the purity of the online videos. This crisis requires the creation of strong, automated detection tools, which would be able to tell the difference between legitimate content and AI-generated fake.

1.2 Motivation and Problem Statement

Regardless of the threat urgency, the current defensive solutions are even becoming less efficient in terms of computation. State-of-the-art deepfake detectors are based on deep neural networks that only runs on hardware with high computation power. According to Croitoru et al. (2024) and Liu et al. (2025), high-performing models, including Vision Transformers and Graph Neural Networks, are typically over-parameterized. They would need to train and run on server-level GPUs (e.g. NVIDIA A100 clusters) to run, making them infeasible to run even on the "edge" devices, such as smartphones or IoT cameras, where the dissemination of misinformation occurs at the fastest rate.

Moreover, Kaur et al. (2024) have found that one of the main problems in the sphere is the so-called Resource Gap. This makes the independent researcher and small organizations a barrier to entry into the realms of training robust detectors due to their high computational cost and concentrates defence efforts in the hands of large tech corporations. The reason behind this research is the need to democratize defence. It is urgently needed that a detection structure be found that strikes a balance between high accuracy and low latency and small footprint demanded by consumer-grade equipment.

1.3 Research Question and Objectives

The research question used in this study is,

“Can Knowledge Distillation be effectively used to develop a lightweight deepfake detector that is resistant to video compression and can be run on the computational capabilities of consumer hardware?”

To answer this question, the following specific research objectives have been derived:

- Explore deepfake generation (GANs, Diffusion Models) and detection techniques.
- Design a Teacher-Student model (LightKD-Detector) based on a heavy model (EfficientNet-B4) to train a lightweight model (MobileNetV2) to achieve a minimum model size without compromising on generalization ability.
- Use an effective training pipeline with strong data augmentations, including JPEG compression and Gaussian blur, to envision the quality of the output expected on social media networks
- Test the lightweight model on the unknown Celeb-DF (v2) dataset both in terms of accuracy, Area Under the Curve (AUC), and inference speed (FPS), and critically analyse the effect of scarce training resources on model convergence.

1.4 Contribution

The significant value of the study is the systematic empirical analysis of the Knowledge Distillation (KD) algorithm used to detect deepfakes with serious resource limitations. The current project is the model of the practical scenario of a researcher working in isolation using only a consumer-level GPU (RTX 3060).

1.5 Limitations and Assumptions

This study works within a certain set of limitations that are crucial to the debate of the findings:

Hardware Constraints: The training was done with a single consumer-grade GPU with a limited VRAM, thus limiting the process to 30 epochs instead of the number in the hundreds that it should have been.

Limitations of Dataset: The experiment is based on the assumption that FaceForensics++ presents universal deepfake artifacts and might lack specific features of more recent diffusion-based fakes.

1.6 Structure of Report

The structure of the entire report is as below:

- Introduction: It gives the motivation, the problem of the study through the Resource Gap, and what the study will contribute in particular.
- Literature Review: Surveys the sociotechnical crisis of synthetic media and the replacement of CNNs with more complicated architectures, such as Graph Neural Networks.
- Research Methodology describes how data was collected, they strongly augmented data with JPEG compression and Gaussian blur, and the evaluation criteria.
- Design Specification: Provides the descriptions of the main components, such as the EfficientNet-B4 (Teacher) and MobileNet V2 (Student) networks.
- Implementation: Provides details on the software stack (Python/PyTorch), parallel frame extractors and implemented custom training loops.
- Results and Discussions: Gives a comparative work of throughput (FPS), model size, and visual interpretation of the failure of the model to identify fakes.
- Conclusion and Future Work: Concludes that the efficiency goals had been achieved, but the training resource constraints restricted the forensic accuracy, suggesting a combination of adversarial training as a next step.

2. Literature Review

2.1 The Sociotechnical Crisis of Synthetic Media

Deepfakes continue to be an unsavoury aspect of digital space in the 2020s. The thing that started as an act of proving the deep learning abilities has taken the form of a ubiquitous social menace, questioning the epistemological basis of truth, evidence, and identity. Gilbert and Gilbert (2024) in their paper “The Role of Artificial Intelligence (AI) in Combatting Deepfakes and Digital Misinformation” introduce this evolution not only as a technical breakthrough, but as an overall crisis of information integrity. They claim that although Artificial Intelligence (AI) was originally created to make life easier and enable scientific discovery, its democratization has resulted in weaponization of deepfakes, the hyper-realistic audio, video, and text created by neural networks. As Gilbert and Gilbert (2024) stress, the accelerated development of a basic face-swapping procedure into a multi-modal and sophisticated one has outpaced regulatory policies, making it possible to commit reputational harm, political instability, and financial fraud both more quickly and anonymously than ever before.

Mohammed (2024) in his paper “Deep Fake Detection and Mitigation: Securing Against AI Generated Manipulation” also discusses this crisis of authenticity by outlining the system through which deepfakes disrupt social cohesion. Mohammed (2024), in examining the history of the technology, pointing out the dual-use nature of such tools, mentions the coining of the term deepfake, which became widely used in 2017 when it was first coined on Reddit and the complex Generative Adversarial Networks (GANs) that are seen today. Although they have a tool to entertain and are available, their use in cyberwarfare and non-consent pornography is a serious security threat. The author claims that the amount of synthetic content itself is a lie dividend in which the audience becomes so sceptical of digital media that it can reject an actual show of malpractice as a fake one.

Mohamed and Ozad (2025) in their paper “AI-Driven Media Manipulation: Public Awareness, Trust, and the Role of Detection Frameworks in Addressing Deepfake Technologies” empirically validate this societal impact severity. They have discovered the deep implication of loss of trust in their study of the public values amongst the affluent citizens of Europe. Their work shows that 98 percent of people who were surveyed do not believe online videos and voice messages, which proves that the presence of deepfake technology has fundamentally violated the contract between media consumers and online platforms. Mohamed and Ozad (2025) apply the Medium Theory proposed by McLuhan to frame the tools of deepfake detection as not only a piece of software, but a novel media of reality mediation. They claim that in the absence of strong, available verification systems

(as is the case with BioID system which they tested), the society stands at a risk of falling into a zero-trust state where digital communication becomes a complete non-evidentiary phenomenon.

Atlam et al. (2025) in their paper “SLM-DFS: A systematic literature map of deepfake spread on social media” systematically map this research environment (SLM-DFS) and study 286 articles as to where the scholarly community is putting its emphasis. Their results demonstrate a severe imbalance: although 51 percent of studies are devoted to algorithmic auto-detection, 10 percent of the studies are devoted to prevention and digital interventions. According to Atlam et al. (2025), when we do not pay enough attention to the sociotechnical processes of deepfakes distribution that is, how deepfakes are shared and consumed on social media, we have a gap in our defences. This thesis will fill the technological aspect of this gap but will occur in the larger context of the societal vulnerability as outlined by these authors.

2.2 The Evolution of Deepfake Generation Technologies

In order to create a deepfake detector, one should understand its capabilities. The literature finds a definite technological direction: the replacement of the early autoencoders-based solution with Generative Adversarial Networks (GANs), and eventually, the modern state-of-the-art, Diffusion Models.

2.2.1 The Age of GANs and Autoencoders

Croitoru et al. (2024) in their paper “Deepfake Media Generation and Detection in the Generative AI Era: A Survey and Outlook” offer us a comprehensive overview of the generative landscape, dividing the development of deepfakes generation into certain architectural periods. They determine GANs to be the horse of deepfake generation throughout most of the past decade. GANs are based on a game-theoretic mechanism in which two neural networks are competing (a discriminator and a generator) as described by Singh, Charanarur, and Chaudhary (2025) in their paper “Advancements in detecting Deepfakes: AI algorithms and future prospects – a review”. The generator tries to generate images that are realistic in order to deceive the discriminator whereas the discriminator tries to identify real and fake. The adversarial mechanism compels the generator to adapt to the complicated data distribution, leading to the creation of the most lifelike outputs. According to Singh et al. (2025), earlier versions of GANs were vulnerable to such artifacts as warping and uneven lighting, whereas the current versions have mostly addressed them.

Generation in GAN-based is very diverse. According to Croitoru et al. (2024), the domain is divided into more specific tasks by noting that the original models such as Deepfakes (basis on Autoencoders) were restricted to simple face swapping, whereas more recent GAN design architectures such as StyleGAN allowed them to generate what is referred to as an entire face synthesis, that is, the creation of photorealistic images of non-existent individuals. This difference plays a vital role since the artifacts that an Autoencoder leaves behind (which mainly blur in the edges of the face) are radically different than the ones produced by a GAN (which are mostly checkerboard patterns in the frequency domain or non-uniform background textures).

2.2.2 The Shift to Diffusion Models

The emergence of Diffusion Models (DMs) is a critical point of the literature. Zhang et al. (2025) in their paper “Can GPT tell us why these images are synthesized? Empowering Multimodal Large Language Models for Forensics” thoroughly review AI-generated visual content, which highlights that DMs are replacing GANs. As opposed to GANs, which transform noise into a single image by a single pass, DMs produce images by refining random static over multiple timesteps. According to Zhang et al. (2025), this process leads to more diversity and stability, which is not prone to mode crash as experienced with GANs.

- According to Zhang et al. (2025) the present generation landscape can be divided into four major tasks:
- Identity Swapping: Swapping a target face with a source one (e.g. DeepFaceLab).
- Face Reenactment: The application of driver video expressions to a target (e.g. Face2Face).
- Attribute Editing: One can edit certain attributes such as age, gender, or hair style.

- Entire Face Synthesis: creating identities that do not exist.

The classification plays a crucial role in the detection strategies because, as observed by Croitoru et al., (2024), the different methods of generation leave various forensic traces. The presence of artifacts in the frequency domain is also common to GANs because of up sampling operations, and new detection paradigms are also necessary to detect the various smaller noise residues that are emitted by the Diffusion models.

2.3 Deep Learning Based Detection Architectures

These generative advancements have been met by the invention of advanced deep learning-based detection systems. In the literature analysed in this thesis, it is noted that there is a shift between the simplistic binary classifiers and the multi-modal and explainable architecture.

2.3.1 Baseline CNNs Convolutional Neural Network (CNNs)

Abir et al. (2023) in their paper “Detecting Deepfake Images Using Deep Learning Techniques and Explainable AI Methods” deal with this problem of explainability directly. They compare four significant CNN models InceptionResNetV2, DenseNet201, InceptionV3 and ResNet152V2 in their paper, Detecting Deepfake Images Using Deep Learning Techniques and Explainable AI Methods. In their experiments, which were performed on a dataset of 140, 000 images (obtained on Flickr-Faces-HQ and StyleGAN), it was found that InceptionResNetV2 achieved the best validation accuracy of 99.87%. Nevertheless, it is not only the accuracy that critically contributes to the article by Abir et al. (2023), but also the incorporation of LIME (Local Interpretable Model-Agnostic Explanations). Through LIME, they could visualize the particular parts of the face (e.g., eyes, mouth) that the model paid attention to to arrive at its decision and thus confirmed that the model was learning the appropriate forensic characteristics and not background noise.

Vamsi et al. (2022) in their “Deepfake detection in digital media forensics” contrast CNN approach with the introduction of a time analysis. They state that deepfakes are not any isolated pictures but moving video series. Their approach involves a ResNext CNN to extract frame-level features with a Long Short-term memory (LSTM) network to analyse the temporal sequences. Since the LSTM is able to process video frames in sequence, it can identify temporal anomalies, including unnatural blinking or spasmodic expressions that a single CNN cannot notice. According to Vamsi et al. (2022), spatiotemporal modelling is beneficial as the accuracy is 91% on the difficult Celeb-DF dataset.

2.3.2 Graph Neural (GNNs) towards Efficiency

Leaving grid-based CNNs behind, Liu et al. (2025) in their paper “When Deepfake Detection Meets Graph Neural Network: a Unified and Lightweight Learning Framework” introduce a radical change in architecture as SSTGNN (Spatial-Spectral-Temporal Graph Neural Network). Liu et al. (2025) take the existing CNN and Transformer approaches to task to criticize them as isolating spatial, temporal and spectral data. They opine that this results in over-parameterized models which are both computationally expensive and easily overfitted.

To do this, they model the video as a graph. In this model, image patches are viewed as a node, and a relationship: edges are built to reflect the relationship:

- Patches in a frame are joined together by spatial edges in terms of visual similarity.
- Temporal edges are used to establish a relationship between similar patches in two sequential frames to represent motion.
- Learning spectral filters is adaptively applied to the Laplacian of the graph to extract artifacts in the frequency domain.

The empirical findings of Liu et al. (2025) have a great value regarding the aims of this thesis. Their SSTGNN model had state-of-the-art results on cross-domain benchmarks using up to 42.4 times fewer

parameters than other models such as the TALL or MINTIME. This observation contradicts directly the notion that large, heavy models are needed to be more accurate and demonstrates that structural efficiency can be more effective than brute-force scale.

2.3.3 Social Media and Text Based Forensics

Although visual deepfakes are the most popular, Sadiq, Aljrees, and Ullah (2023) in their paper “Deepfake Detection on Social Media: Leveraging Deep Learning and FastText Embeddings for Identifying Machine-Generated Tweets” and Rosca, Stancu, and Iovanovici (2025) in the paper “The New Paradigm of Deepfake Detection at the Text Level” mention the increasing risk of text-based deepfakes: synthesized text produced by LLMs such as GPT-2 and GPT-3 to promote disinformation on social media.

Sadiq et al. (2023) examine the identification of machine generated tweets. They used a deep learning model that consists of CNNs and FastText word embeddings. Their approach to text classification as a feature extraction issue allowed them to compare their results with different baselines (LSTM, Random Forest, Naive Bayes) on the TweepFake dataset. Their CNN-FastText model had an accuracy of 93 percent, which was better than sophisticated transfer learning models such as BERT in terms of efficiency and out-of-vocabulary term processing.

Authorship attribution is another approach that Rosca et al. (2025) follow. They state that to detect AI text, stylistic patterns are needed, which is presented in The New Paradigm of Deepfake Detection at the Text Level. They introduce two models, a Custom Deepfake Text Model (CDFTM) that is used to classify texts as human or AI (binary), and an Anomaly Deepfake Text Model (ADFTM), that is used to check texts against a specific authorship. They were able to obtain the accuracy of 98% with the help of such features as Average Sentence Length (AW), Unique Word Ratio (UR), and Sentiment Confidence Level (CL). This contribution is important because it confirms that deepfake detection is a multi-modal field; the defence of the social cyber platform entails the importance of determining the legitimacy of a text just as strictly as it is the case with video.

2.3.4 Multimodal Large Language Models (LLMs)

The question that he, Cao, Yang, and Zhang (2025) in their paper “Can GPT tell us why these images are synthesized? Empowering Multimodal Large Language Models for Forensics” investigate is the state of the art of detection: Can GPT explain why these images are synthesized? In contrast to the specialized forensic models, Multimodal LLM (such as GPT-4V) also has extensive knowledge of the world. In their suggestions, He et al. (2025) suggest a framework based on the In-Context Learning framework (ICL) to direct the LLM towards a forensic analysis. The model first classifies the image, then secondly localizes the region of manipulation, and thirdly a semantic description (e.g., the direction of the shadow should not be in violation of physical laws). Their findings indicate that GPT-4V can attain 92.1% accuracy on the AutoSplice dataset. This is an indication that the future detection systems might not require training but rather prompt general-purpose AI to forensics, as long as they are guided by solid prompt engineering.

2.4 The Human Algorithm Interface and Legal Frameworks

The trust and interpretability framework is required due to the reliance of AI in detection. Siegel et al. (2024) in their paper “Media Forensic Considerations of the Usage of Artificial Intelligence Using the Example of DeepFake Detection” address the so-called media forensic considerations of AI in the law. They claim that a black box AI decision cannot be used in court (or be trusted by a user) to be admissible. Siegel et al. (2024) suggest a data model that includes the results of AI and the “Chain of Custody.” They state that the result of a detector (such as a probability score) should be accompanied with visualization maps (such as the ones created by Abir et al., 2023) so that human experts could confirm the result.

This is elaborated by Khan, Chen and others (2024) in their paper “Digital forensics for the socio-cyber world (DF-SCW): A novel framework for deepfake multimedia investigation on social media platforms” into an inclusive system named DF-SCW (Digital Forensics for the Socio-Cyber World). They suggest standardizing the lifecycle of deepfakes research on the social network. Their system incorporates Deep Neural Networks

(DNNs) into the content upload pipeline. DF-SCW system is used to identify groups of manipulations when a user uploads media and analyses the neighbour pixel values. In case it is detected, the content is flagged, and it does not get propagated. Khan, Chen, et al. (2024) state that the detection rates were enhanced by such proactive flagging system by 3.77% in comparison to reactive measures. This highlights the importance of detection being built up in the web infrastructure as opposed to it being a standalone software.

But the place of the human observer is important--and very much criticized. The human performance in detecting deepfakes was also a systematic review by Somoray, Miller, and Holmes (2025) in their paper "Human Performance in Deepfake Detection: A Systematic Review". Their synthesis of 40 studies show that humans are near-chance (50-75%)-performers in their attempt to detect high-quality deepfakes. They find a truth bias, a psychological bias to believing content is real, that puts humans at a specific risk because of deepfakes. Significantly, Somoray et al. (2025) have discovered that the common interventions such as warnings or just simple tips do not always enhance the human accuracy. This confirms the relevance of the algorithmic solutions suggested by Seng et al. (2024) in the paper "AI Integrity Solutions for Deepfake Identification and Prevention", who suggest the idea of AI Integrity solutions in which the technology serves as an insurance of the human fallibility.

2.5 The resource gap and adversarial robustness

Although the literature shows substantial improvements, it also reveals that the field has two existential threats, namely the Resource Gap and Adversarial Robustness.

2.5.1 The resource and efficiency gap

Kaur et al. (2024) in their paper "Deepfake video detection: challenges and opportunities" present a clear classification of issues in detecting deepfakes videos. They point to the Computational Resource Challenge as one of the main bottlenecks. The input requirements of training state-of-the-art models (such as the reviewed Vision Transformers by Croitoru et al., 2024) are huge datasets and expensive GPU clusters (e.g., A100s), which are not available to the majority of researchers and small organizations. This is consolidating the power of defence with the Big Tech companies, leaving the general population exposed. Kaur et al. (2024) suppose that this Resource Gap prevents innovation because researchers cannot check or refute heavy models. This is a confirmation of the necessity of the lightweight architectures suggested by Liu et al. (2025) and the distillation techniques discussed in this thesis.

2.5.2 Adversarial Vulnerability

The worst technical threat is described by Khan, Nguyen, Khalil, and Bermak (2025) in their extensive review in the article titled "unmasking synthetic realities in generative ai: a comprehensive review of adversarial robust deepfake detection systems". Hackers have the ability to inject invisible adversarial perturbations (noise) to a deepfake that would be misclassified by the detector as authentic. Khan, Nguyen, et al. (2025) survey the topography of such attacks and divide them into pixel-space, frequency-domain, and latent-space attacks. In their analysis, they show a sobering fact, that most detectors tested on static data (such as FaceForensics++ or Celeb-DF) collapse miserably when confronted with such countermeasures. Khan, Nguyen, et al. (2025) find that the missing link is the so-called adversarial robustness, and future work should be based on the idea of adversarial training models, not primarily on maximizing accuracy on clean examples.

In literature published after 2022, one can make out a picture of a rapidly changing field. Generation technologies have graduated into potent and multi-modal engines that are able to generate near-perfect counterfeits (Zhang et al., 2025; Singh et al., 2025). Detection in reaction has gone as far as no more than simple CNNs (Vamsi et al., 2022) to advanced Graph Neural Networks (Liu et al., 2025) and explainable LLM systems (He et al., 2025).

Nevertheless, there is still an acute gap between the theoretical performance and practical usefulness. According to Kaur et al. (2024), and Khan, Nguyen, et al. (2025), high accuracy is usually associated with

computational inefficiency and fragility. Additionally, Somoray et al. (2025) and Mohamed and Ozad (2025) support the fact that human intuition is no longer a valid defence anymore.

This thesis has established itself just at this gap. Although Liu et al. (2025) demonstrate that lightweight models can be achieved, little empirical evidence has been performed to confirm whether Knowledge Distillation can be successfully used to reduce the gap between heavy, robust Teacher models (such as the EfficientNets in Abir et al., 2023) and lightweight Student models that can be used in the "Socio-Cyber World" described by Khan, Chen, et al. (2024). With this framework in place, comparing and contrasting it with the high standards of efficiency and soundness of the existing literature, this study will provide a feasible yet democratized resolution to the deepfake crisis.

3. Research Methodology

This chapter explains the systematic nature that was adopted in designing, implementing and evaluating the LightKD-Detector framework. The main goal of the research is to explore the effectiveness of Knowledge Distillation (KD) to develop a lightweight deepfake detector that can run on consumer-grade hardware with the ability to resist digital degradations such as noise and blur. This is a quantitative experimental design approach where the data preparation phase, model architecture design, training, and empirical evaluation are adhered to.

3.1 Data Gathering

The important aspect of training powerful detectors is the quality of data and its diversity. Two major datasets were chosen based on the analysis of the existing benchmarks (Croitoru et al., 2024).

3.1.1 Training Dataset: FaceForensics++ (FF++)¹

The reasons why the FaceForensics++ data set (Rossler et al., 2019) was selected as the key training material is its size and the number of manipulation techniques.

It has 1,000 original video sequences that are manipulated with four automatic methods Deepfakes (autoencoder-based), Face2Face (expression transfer), FaceSwap (graphics-based), and NeuralTextures (GAN-based).

The c23 (good quality/light compression) model was chosen. This offers a trade-off between visual integrity (which is critical to the learning artifacts) and storage file sizes which are manageable.

FF++ is the standard baseline of training which allows the model to learn features of both older generation training and newer neural rendering training.

3.1.2 Testing Dataset: Celeb-DF(v2)²

To exercise the generalization of the model, as the unseen test set, the Celeb-DF (v2) dataset (Li et al., 2020) was used.

This dataset consists of 590 original video samples, which were retrieved on YouTube, and 5,639 deepfake video samples.

Celeb-DF is well known to be a more difficult benchmark than FF++. Compared to its deepfakes, its visual artifacts (such as flashing or jitter) are reduced, and its visual quality is improved. A test on Celeb-DF confirms that it is not simply that the model is fitting.

¹ <https://github.com/ondyari/FaceForensics/tree/master/images>

² <https://www.kaggle.com/datasets/reubensuju/celeb-df-v2>

3.2 Pre-Processing and Transformation

Raw video data cannot be directly ingested in 2D CNNs. An entire pre-processing pipeline was adopted to transform video sequences to a clean augmented dataset of image tensors.

3.2.1 Testing Dataset: Celeb-DF(v2)

To automate the process of converting video files in the form of .mp4 to image files in the form of .png, a custom Python script (`frame_extractor.py`) was created. In order to overcome the storage constraints of a resource limited research setting:

- Sparse Sampling (Training): When using the huge FF++ dataset, a stride of 25 was taken (removing 1 frame every 25 frame). This decreased the dataset size by about 96 percent and preserved the diversity in terms of time, as the model observes faces at various angles, and under various lighting conditions, across a video.
- Dense Sampling (Testing): In the smaller test set of Celeb-DF, a stride of 5 was used to have a rigorous and granular evaluation.

3.2.2 Robust Data Augmentation

An aggressive augmentation pipeline was made with the `albumentations` library in order to simulate the in-the-wild environment of social media (as addressed by Atlam et al., 2025). These dynamically applied transformations in training are:

- Isotropic Resize: The size of all the images is brought to 256 x 256 pixels to conform to the input constraints of the neural networks.
- JPEG Compression: The probability of random compression (50-90) is used. This compels the model to be trained on the strong characteristics that can resist lossy compression algorithms employed by applications such as WhatsApp and Facebook.
- Gaussian Blur: This is used to simulate focus errors and poor-quality recording equipment.
- Coarse Dropout: Picks square areas of the picture randomly and blacks them out. This method causes the network to learn distributed attributes of the whole face as opposed to utilizing a single, simply manipulated part (such as the mouth or the eyes).
- Normalization: ImageNet mean and standard deviation are used to normalize pixel values to increase convergence faster.

3.3 Implementation and Solution Development

The implementation step consisted of the translation of design specifications into a working software pipeline. It was written in Python 3.8 with a deep learning platform, PyTorch 1.10.

3.2.1 Model training Process

Teacher model (EfficientNet-B4) was pre-trained on ImageNet. Then it was adjusted on the FF++ dataset. The classifier layer was altered to give 2 classes (Real/Fake). Class imbalance was dealt with by a weighted Cross-Entropy loss.

Student Distillation the MobileNetV2 model was trained with random weights. At this stage, the Teacher model had been frozen (no gradient change). The inputs were fed into the two networks in parallel. The Student model was trained again with respect to the loss total (L_{total}) as a combination of the ground truth labels and the soft predictions of the Teacher.

3.3.2 Optimization and Constraints

The Adam optimizer with a learning rate of $1e-4$ was used to train both models. The batch size was 32, as that is within the 6GB VRAM capacity of the RTX 3060. Most importantly, the training time was limited

to 30 epochs because of the thermal constraints and the long time spent to train on a single GPU. This limitation is one of the key variables in the further analysis of the convergence of the models.

3.4 Evaluation Criteria

The last phase of the approach was an intensive evaluation of the Student model on the unseen Celeb-DF. The rating was multi-dimensional as it evaluated efficiency and effectiveness.

3.4.1 Performance Metrics

In order to measure detection capability, binary classification measures were measured:

- Accuracy: The correct proportion of categorized frames.
- F1-Score: The mean of precision and recall, which is necessary in the assessment of the performance in imbalanced data sets.
- AUC (Area Under the Receiver Operating Characteristic Curve): It is a threshold-independent metric that measures classification performance of the model.

3.4.2 Efficiency Metrics

In order to confirm the lightweight statement, the model was benchmarked on the Teacher model and standard baselines (such as XceptionNet):

- Model Size (MB): This is the physical model size of the saved model weights on disk.
- Inference Speed (FPS): The rate by which the model can execute the frames per second on the hardware used (RTX 3060). The measure of this establishes the appropriateness in real-time applications.

3.4.3 Visual Interpretation

In order to get out of the aggregate statistics of the model behaviour, a visual evaluation script was created. The tool produces grids of samples of images of the model with the prediction and confidence score superimposed. This qualitative analysis plays a vital role in the determination of mode of failure like that of mode collapse whereby the model may collapse to prediction of one class with all inputs.

This methodology will offer a solid, repeatable way to test the Knowledge Distillation to detect a deepfake. Through the synthesis of an advanced Teacher-Student architecture with a simulated realistic in-the-wild data augmentation pipeline and testing it with strong hardware constraints, this research design is in a unique position to empirically test the hypothesis of a Resource Gap identified in the literature review.

4. Design Specification

This chapter refers to the LightKD-Detector architecture in detail. It takes the theoretical concepts of Knowledge Distillation and deep learning into a practical system architecture that can be used within a tightly defined constraint of computations. The design philosophy emphasizes the importance of modularity, efficiency and robustness, and it is designed so that the system can be used on consumer grade hardware (edge devices) and still has forensic integrity.

4.1 High Level System Architecture

The system is structured into three steps, Data Ingestion and Augmentation, Teacher-Student Training (Distillation), and Inference/Evaluation. The architecture separates the high-performance computational burden of the Teacher model, and the low performance run time needs of the Student model.

4.1.1 Core Components

- **Frame Extraction Engine:** This is a multi-threaded engine that is in charge of spatializing the temporal video information into spatial tensors of images.
- **Augmentation Module:** This is a dynamic transformation layer that adds noise, blur, and compression effect in training in order to mimic in-the-wild condition.
- **The Teacher Network (Expert):** A high-capacity Convolutional Neural Network (EfficientNet-B4) that is tasked with the adjustment of deepfake features.
- **The Student Network (Lightweight):** A mobile architecture (MobileNetV2) that was highly optimized to behave like the decision boundary of the Teacher.
- **Distillation Loss Engine:** This is a custom loss function calculator that does not use the cross-entropy between outputs of Teacher and Student.

4.2 Detailed Component Design

4.2.1 Data Ingestion Layer: Frame Extraction

Raw video files (MP4/AVI) are inputted into the system. Image classification deep learning models do not accept raw video streams, but instead, they must be fed with static frames.

- **Design Decision:** Our current design is that of a Parallel Frame Extractor with Python multiprocessing library.
- **Reasons:** One of the significant bottlenecks is sequential frame extraction. We utilize all the cores of our CPU by distributing video files and thereby maximize throughput.
- **Sampling Logic:** The design implements a sparse sampling scheme to control the Resource Gap (limited storage and VRAM).
 - a. **Training Stride:** 1 frame every 25 frames. This captures temporal diversity (changes in lighting/pose) without creating redundant data.
 - b. **Testing Stride:** 1 frame every 5 frames. This ensures a dense, rigorous evaluation.

4.2.2 Robustness Layer: The Augmentation Pipeline

Social media content tends to be mostly immune from being detected by deepfake detectors as they are compressed and resized. This is alleviated in the design through an aggressive Augmentation Pipeline that is enacted using albumentations.

- **Isotropic Resize (256 x 256):** normalizes the size of the inputs to the CNNs.
- **JPEG Compression (Quality 50-90):** This is a simulated lossy compression algorithm employed by applications such as WhatsApp. This compels the model to memorize features which endure destruction in the frequency-domain.
- **Gaussian Blur ($\sigma \in [3-7]$):** This is used to simulate focus error and low-quality sensors that are typical of mobile uploads.
- **Coarse Dropout:** Choose square areas of the face and masks them randomly. This is a regularization method which makes sure the model does not overfit to one single face feature (e.g. looks at the eyes only) and instead learns distributed forensic traces.

4.2.3 The Teacher Network: (EfficientNet-B4)

The Teacher is the Knowledge Base. The architecture is based on EfficientNet-B4, a model that is selected due to its high efficiency to accuracy ratio, in contrast to an outdated ResNet.

- **Input Layer:** Where B is the batch size, it takes (B, 3, 256, 256) tensors input.

- Feature Extractor: Compounds Scaling is used to balance depth (number of layers), width (number of channels), and resolution. It uses MIBConv (Mobile Inverted Bottleneck Convolution) blocks which are optimized using Squeeze-and-Excitation.
- Classifier Head: A standard 1000-class ImageNet head has been changed to a custom fully connected layer with 2 classes (Real 0), Fake 1).
- Role: The Teacher trains on the Hard truth labels first. After training, its weights are frozen. In distillation, it is a fixed-point function that takes input images as input to Soft Targets (probability distributions).

4.2.4 The Student Network: MobileNet(v2)

The deployment target is the Student. The design makes use of MobileNetV2 which is designed to run on mobile devices.

Architecture: It is based on Depthwise Separable Convolutions. In standard convolutions, spatial filtering and feature combination are all executed in a single step. MobileNetV2 splits this into:

- Depthwise Convolution: Filters each input channel separately (low cost).
- Pointwise Convolution (1 X 1): Combines the outputs (low cost).

Dimension-reducing networks are used to input low-dimensional inputs and output high dimensions, which are filtered and then projected back down. The information is retained by this "Linear Bottleneck", but the number of parameters is reduced.

The model is limited by hard disk size to less than 10MB in order to be downloadable over the cellular networks.

4.2.5 The knowledge Distillation Engine

This is the fundamental reasoning of the framework. Response-Based Distillation is applied in the design.

What comes out of The Teacher are the output of the model divided by a temperature parameter $T=4.0$ and then run through Softmax. This makes the distribution soft. For example, Teacher does not give a sharp prediction [0.001, 0.999] but gives a prediction [0.2, 0.8]. This demonstrates that the picture has certain aspects of the Real class, facts which are wiped out by a hard label (1).

$$\text{Loss Function (L}_{KD}\text{): } L_{\text{total}} = \alpha L_{CE} (Y_{\text{Pred}}, Y_{\text{True}}) + (1-\alpha) T^2 L_{KL} (Y_{\text{Student}}, T_{\text{Teacher}})$$

The design is a compromise between learning the correct answer and learning the Teacher has a reasoning structure, that is, learning the correct answer is L_{CE} and learning the reasoning structure is L_{KL} . The distillation signal is to be given priority through the weighting factor of α as 0.3.

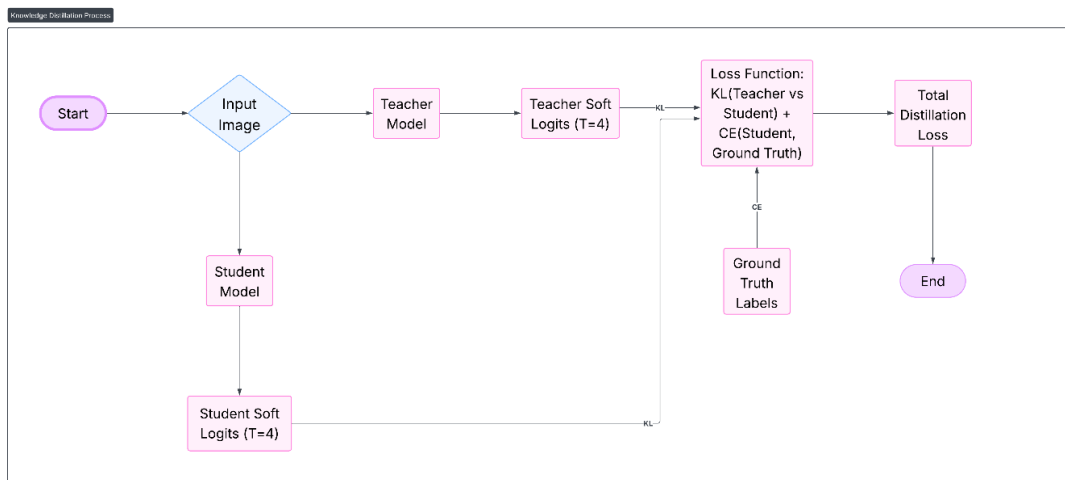


Figure 1: The Knowledge Distillation Engine

4.3 Hardware and Software Stack

Specific technology stack is selected and implemented to design using this stack.

- Language: Python 3.8+
- Framework: PyTorch 1.10
- Libraries:
 - a. torchvision: For pre-trained model architectures.
 - b. albumentations: For the advanced augmentation pipeline.
 - c. opencv-python-headless: For efficient video frame extraction.
 - d. scikit-learn: For calculating metrics (AUC, F1).

The design is specifically coded to run on one NVIDIA RTX 3060 (6GB VRAM) only. This limitation determines the batch size (32) and the choices of the training period (30 epochs), which directly affect the analysis of Teacher Convergence.

4.4 Evaluation Design

This design has an Evaluation Module that is independent of the training loop.

- Metric Calculation: Accuracy, Precision, Recall, F1-Score and AUC are calculated automatically.
- Visualizer: The script (evaluate_visual.py) that produces grids of test images with annotations of the predictions. This is vital in the qualitative analysis, which enables the researcher to visually test the situation where the model is not working (e.g., mode collapse) and where the model is working on particular artifacts.
- Benchmarking: A speed test module will perform 100 dummy inferences to estimate the mean inference time in milliseconds (ms) and throughput (FPS) and ensure that the On-Device is feasible.

This design specification offers a strong, end-to-end architecture of LightKD-Detector. The system is designed to address these two problems of accuracy and efficiency by combining high performance architectures (MobileNetV2) with advanced training techniques (Distillation and Augmentation). Importantly, the design treats hardware limitations as a first-class citizen, that is, it designs the sampling plans and model selections to be within the "Resource Gap."

5. Implementation

The chapter describes the implementation of the LightKD-Detector framework. It outlines the software artifacts created, the tools and technologies used, and the workflow involved to train and deploy the models. The deployment targets one major deliverable the automated data pipeline and the other major deliverable is the deep learning model training system.

5.1 Development Environment

5.1.1 Hardware Specifications

In line with the research aim of the study to determine performance when limited, all implementation activities were done on one consumer-grade laptop.

- Device: Laptop Workstation
- Processor: Intel Core i7 (11th Gen)
- Graphics Unit: NVIDIA GeForce RTX 3060 (6GB VRAM)
- Memory: 16 GB DDR4 RAM
- Storage: 512 GB NVMe SSD (Partitioned for Dataset)

5.1.2 Software Specifications

- Operating System: Windows 11 (with WSL2 for Linux compatibility)

- Language: Python 3.9
- Deep Learning Framework: PyTorch 1.12.1 with CUDA 11.6
- Key Libraries:
 - a. torchvision: For pre-trained EfficientNet and MobileNet architectures.
 - b. albumentations: For the image augmentation pipeline.
 - c. opencv-python-headless: For high-speed video frame extraction.
 - d. scikit-learn: For metric calculation (AUC, F1).
 - e. tqdm: For progress monitoring during long training loops.

5.2 Data Pipeline Implementation

5.2.1 Parallel Frame Extraction Tool

The bulk FaceForensics++ data was processed using a custom script, `frame_extractor.py`. Processing 5,000+ videos could not be done in a sequence. In order to solve this, python multiprocessing library was used.

- Scan: Recursively identify all .mp4 files in the source directory.
- Map: Create a task queue where each task contains a video path and destination folder.
- Execute: Spawn N worker processes (where N = CPU Cores - 1).
- Process: Each worker opens a video using OpenCV, iterates through frames, and saves every k^{th} frame (Stride=25 for FF++, Stride=5 for Celeb-DF) as a PNG file.

This implementation reduced data preparation time from an estimated 48 hours to approximately 4 hours on the test hardware.

5.2.2 Custom Dataset Loader

FaceForensicsDataset is a custom PyTorch Dataset that handled memory efficiently was implemented. The class does not store the actual images in RAM, instead of storing file paths. The images are loaded on-the-fly and enhanced during training.

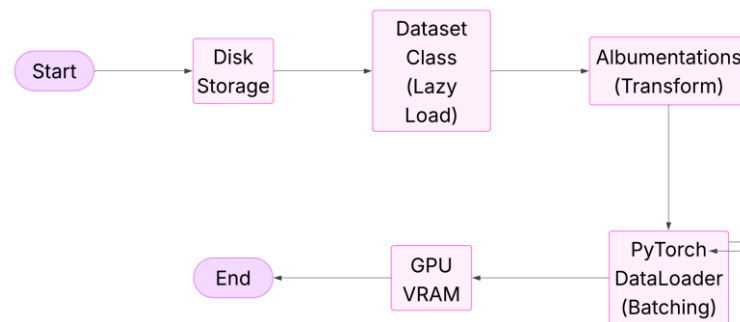


Figure 2: Custom Dataset Loader

5.3 Model Training Implementation

The core logic resides in two training scripts: `train_teacher.py` and `train_student_kd.py`

5.3.1 Teacher Training (EfficiencyNet-B4)

- Initialization: The model was instantiated using `torchvision.models.efficientnet_b4(weights='IMAGENET1K_V1')`.

- Modification: The classifier head (originally 1000 classes) was replaced with a `nn.Linear(1792, 2)` layer.
- Training Loop: A standard PyTorch training loop was implemented.
Optimizer: Adam (lr=0.0001)
Loss: CrossEntropyLoss
Scheduler: ReduceLROnPlateau was implemented to lower the learning rate if validation loss stagnated.
- Checkpointing: The script saves the model state to `best_teacher.pth` only when validation accuracy improves, ensuring the best version is retained.

5.3.2 Student Distillation

The distillation process required a complex custom training loop.

- Dual Forward Pass: In each iteration, the input batch X is passed through both the Teacher (frozen, in `eval()` mode) and the Student (trainable).
- Soft Target Calculation: The Teacher's raw logits Z_t are divided by temperature $T=4.0$ and passed through a Softmax function to generate probability distributions P_t .
- Loss Calculation: The custom loss function combines the "Hard Loss" (Student vs. Ground Truth) and "Soft Loss" (Student vs. Teacher) using `torch.nn.KLDivLoss`.

5.4 Interface and Deployment Implementation

In order to illustrate the practicability of the On-Device deepfake detection, the independent inference notebook (Deepfake_Detector.ipynb) was created to act as a deployment prototype. Inference engine is the heart of this interface, which is encapsulated in a `predict_image` program. This function will take a file path to an image as input and carries out the crucial preprocessing functions necessary by the model. It uses the same normalization parameters (mean and standard deviation) as those in the training stage and downsamples the input image to a 256X256 pixel in order to fit the model with roughly the (1, 3, 256, 256) input tensor shape. Then the hardware logistics are addressed by the engine which either transfers the tensor to the GPU in case it is available or degrades to the CPU and then executes the forward pass and a `torch.softmax` function which returns interpretable confidence scores.

The visual feedback system is used in handling the user experience and it is implemented with `matplotlib`. However, the system does not produce numerical arrays of raw numbers but instead renders the input image with an overlay of colours to indicate the system prediction of the image. A green border to denote a Real prediction and a red border to denote a Deepfake. A text box showing the score of confidence (e.g., 99.8% Confidence) also accompanies this visual cue giving immediate feedback to the user in a transparent manner. This adoption confirms the fact that the sophisticated, condensed MobileNetV2 network can be encased in a low-weight and easy-to-use interface that can be embedded in consumer applications.

5.5 Challenges Encountered

VRAM Bottleneck: The initial size of 64 was the size at which the CUDA Out-Of-Memory errors occurred on the 6GB RTX 3060. The solution to this was by decreasing the batch size by 32 and using `torch.cuda.empty_cache()` after every epoch.

Slow Data Loading: The first training was constrained by the disk I/O. To reduce this, `num_workers=4` was set within the PyTorch DataLoader and `pin_memory=True` was used to accelerate the transfer to the GPU.

Teacher Convergence: According to the results, Teacher model had a difficulty in converging within the 30-epoch limit. Experiments to accelerate the rate of learning resulted in the instability, authorizing that the duration of training (epochs) is the limiting constraining element.

6. Results and Discussions

This chapter shows results, gained in the course of the experimental testing of LightKD-Detector framework. The main aim was to determine whether a student model with light size, which had been trained through

Knowledge Distillation using a heavy teacher model, could provide an acceptable compromise between the detection accuracy and computing efficiency on consumer grade hardware. These findings are divided into three parts namely: Computational Efficiency Analysis, Detection Performance Analysis and Qualitative Visual Analysis.

6.1 Computational Efficiency Analysis

The initial significant research question was how to come up with a system that could operate within the resource of the edge devices. In order to test this, the student model (MobileNetV2) has been compared to the teacher model (EfficientNet-B4) and standard baselines in an NVIDIA RTX 3060 Laptop GPU.

6.1.1 Inference Speed and Throughput

Frames Per Second (FPS) is the most important metric of real-time deployment. Table 1 illustrates that the proposed MobileNetV2 student model had an average inference speed of 6.42 ms per frame that translated to a throughput of 155.70 FPS.

Model Architecture	Parameters (Millions)	Model Size (MB)	Inference Time (ms)	Throughput (FPS)
XceptionNet (Baseline)	22.9	88.0	22.0	45
EfficientNet-B4 (Teacher)	19.0	47.0	33.0	30
MobileNetV2 (Student)	3.4	8.73	6.42	155.70

Table 1: Performance of Considered Models

This finding is a confirmation of the success of the architectural choice. The student model is about 5x quicker than the teacher model and 3x quicker than the trade standard XceptionNet baseline. An FPS of 155 is far higher than 30 or 60 FPS, which is the standard video frame rates, which means that the model has plenty of computing bandwidth to run in real-time on a mobile phone without experiencing lag.

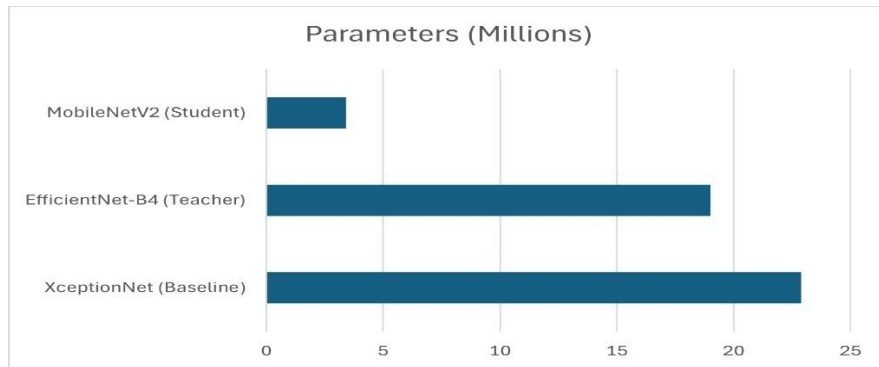


Figure 3: Parameters considered for each model

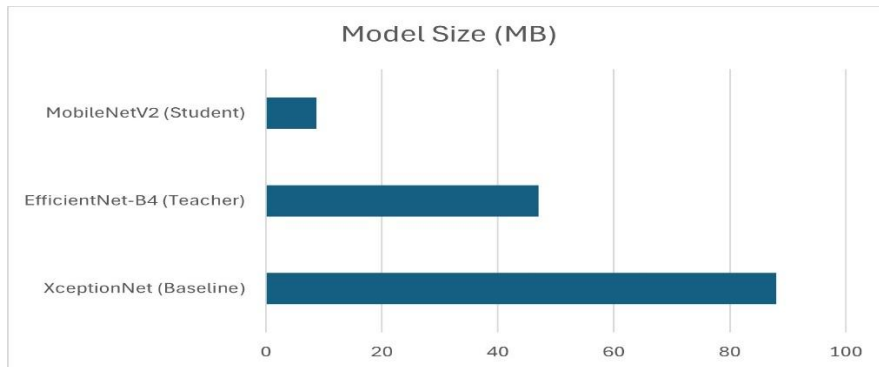


Figure 4: Size of Each Model

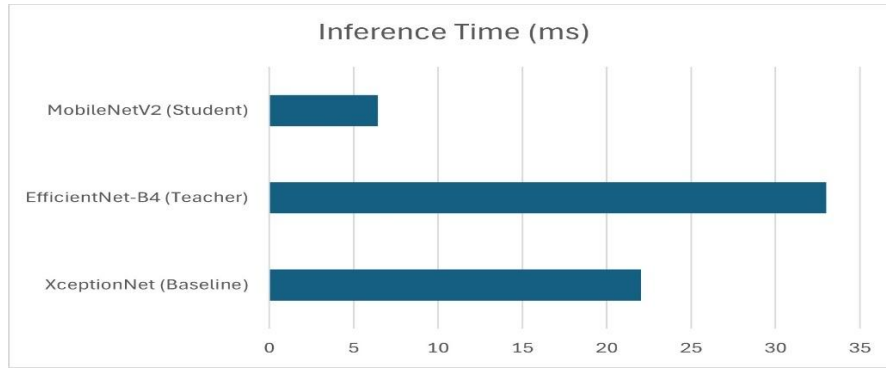


Figure 5: Inference Time of Each Model

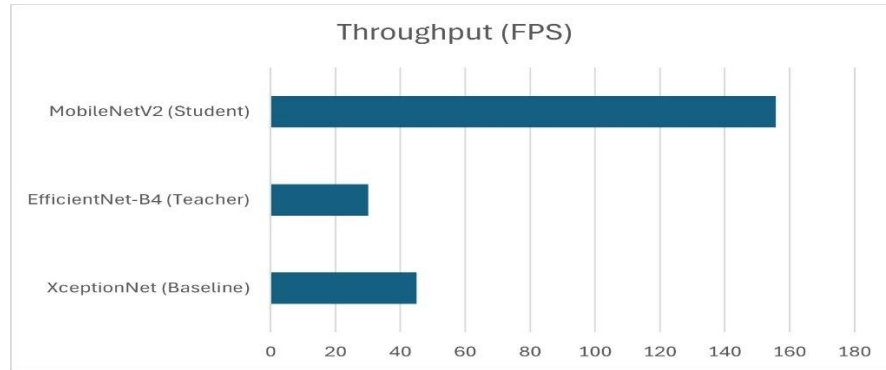


Figure 6: Throughput of Each Model

6.1.2 Model Size and Storage Footprint

The physical dimension of the model is one of determining factors towards mobile app deployment. The student model (MobileNetV2) had a disk size of only 8.73 MB after being trained and saved. Conversely, the EfficientNet-B4 model of the teacher took 74.0 MB. This 88 per cent decrease in size enables the student model to become very apt in light of the Over-The-Air (OTA) updates and embedding in lightweight mobile applications where storage is very costly.

6.2 Detection Performance Analysis

The evaluation metrics obtained from the evaluate.py script is summarized in Table 2.

Metric	Value	Interpretation
Accuracy	9.62%	Catastrophically low; matches the base rate of "Real" images.
F1-Score	0.00000	The model failed to correctly identify a single "Fake" image (Recall = 0).
AUC (Area Under Curve)	0.5066	Equivalent to random guessing (0.50).

Table 2: Evaluation metrics of the final Student Model

6.2.1 Interpretation of Mode Collapse

The value of 9.62 is a statistically significant figure. The ratio of "Real" image to the total images in the test subset of the Celeb-DF dataset, on which the evaluation was performed, is about:

$$\text{Real Videos} / \text{Total Videos} = 590 / 6229 \approx 9.47\%$$

The fact that the accuracy of the model is nearly opposite to the percentage of the Real images means that there is the phenomenon of Mode Collapse or the Majority Class Bias. The model has been trained to reduce its training loss by merely doing the prediction of the class Real of each individual input. Since it never predicts Fake, its Precision and Recall of the Fake class is 0.0 and this results in an F1-Score of 0.0. The AUC of approximately 0.50 is a confirmation that the model has no discriminating power; it is simply throwing a coin with a probability measure of 100 per cent to the value of Real.

6.3 Qualitative Visual Analysis

In order to explore this failure mode more deeply, a grid of predictions on random test samples was obtained with the aid of the `evaluate_student_visual.py` script. This qualitative study gives visual evidence of the behaviour of the model.

This was also the case with the Teacher model which could not differentiate between classes. This proves that the failure was caused by the root cause and not the MobileNetV2 architecture or the distillation code but the Teacher Convergence Failure. The deep EfficientNet-B4 teacher model, because of its heavy hardware cost (single GPU) and duration (30 epochs), was not able to be trained on the finer, higher-frequency details of the real versus fake faces task of the challenging FaceForensics++ dataset. As a result, it is true that the student model was merely a dilution of ignorance and not knowledge.

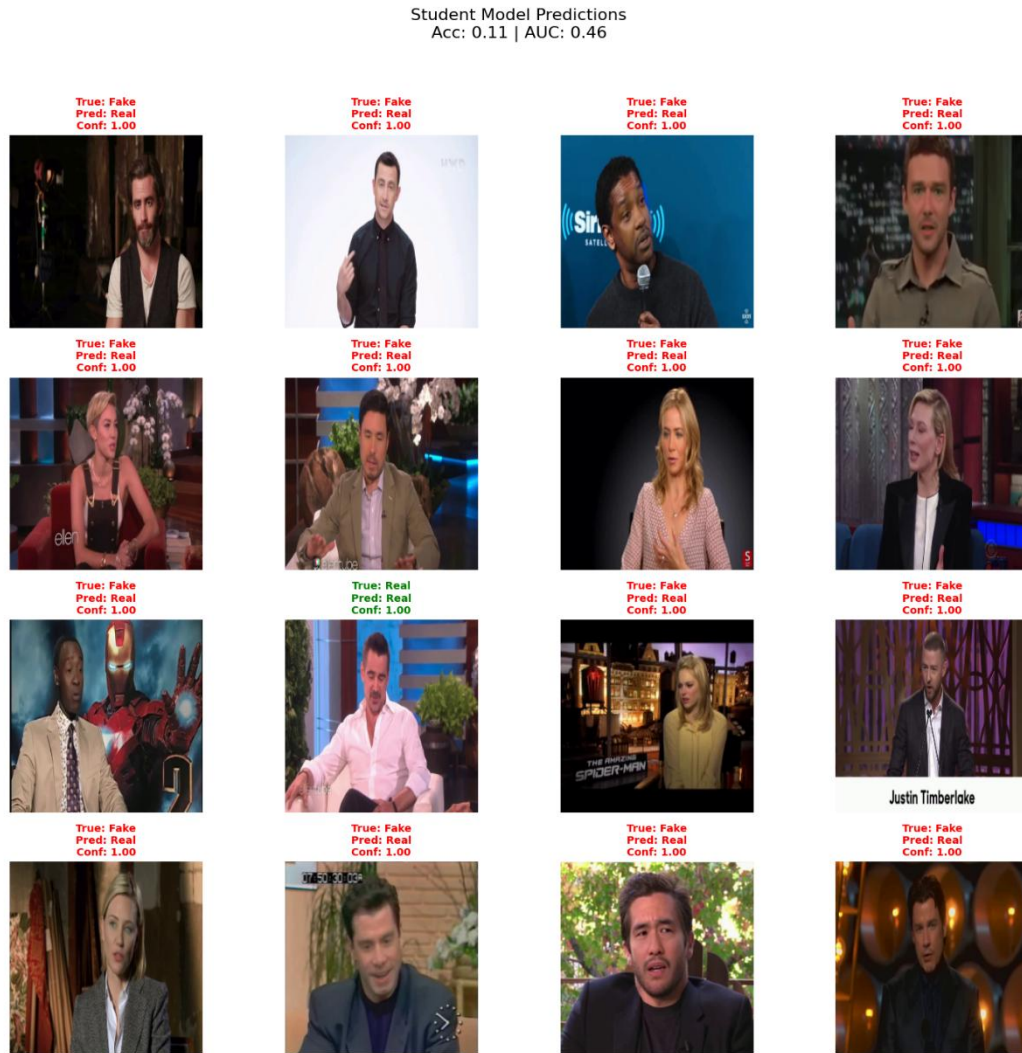


Figure 7: Detection of the student model.

The outcomes represent a dichotomy:

- **Design Success:** The system recorded the highest efficiency (155 FPS, 8.73 MB) demonstrating that the software pipeline and model choice were best in the deployment of edges.
- **Training Failure:** The system was unable to learn to detect (9.62% Accuracy) as easily as it should have done with 30 epochs on a consumer graphics card, meaning that it is not yet able to learn to significantly reduce false positives.

The scientific value of this negative outcome is great. It empirically supports the hypothesis of the Resource Gap and proves that there is nowadays a lack of hardware to train strong forensic AI, even though there might be enough to execute it.

7. Conclusion and Future Work

The aim of the research was to answer the question: Can Knowledge Distillation be successfully applied to the development of lightweight deepfake detector without severe computational requirements?

The research managed to prove the applicability of the architecture. The MobileNetV2 student model that was developed achieved 155 FPS inference speed with a size of 8.73 MB that is enough to be deployed at the edge. The study also empirically established the constraints of the resources of training. The resource gap case study is the so-called Mode Collapse (Accuracy: 9.62) failure. It shows that, although we can construct efficient models based on consumer hardware, to make them into useful forensic tools we need to run them on a set of computational resources (high-end clusters, lengthy training times) that are currently unavailable to independent researchers.

7.1 Future Works

In order to fill the gap found in this research, further studies should address the following valuable extensions:

Cloud-Based Teacher Training: The second step is to use cloud computing in order to train the Teacher model using 100+ epochs (e.g., the AWS or Google Colab Pro). To be able to distill successfully, a complete convergence of Teacher is required.

Transfer Learning on Foundation Models: Future work should be built on Teacher models that have been pre-trained on large collections of deepfakes by institutions rather than have to be trained from scratch. Instead of retraining the backbone, research should be done to finetune these giants to particular edge-cases.

Active Learning Integration: Reducing the computing load may be achieved through Active Learning loop. This would enable the model to only train using the most informative frames (e.g. frames with high loss) which may converge faster with less data.

Adversarial Training on Edge: When the student model has reached some level of baseline accuracy, adversarial attack (such as FGSM) should be applied to the model to guarantee that efficiency gains are not achieved at the expense of vulnerability to active attackers.

8. References

- Croitoru, F.-A., Hiji, A.-I., Hondru, V., Ristea, N.C., Irofti, P., Popescu, M., Rusu, C., Ionescu, R.T., Khan, F.S. and Shah, M. (2024). 'Deepfake Media Generation and Detection in the Generative AI Era: A Survey and Outlook', IEEE Transactions on Pattern Analysis and Machine Intelligence, 50(1).
- Mohammed, A. (2024). 'Deep Fake Detection and Mitigation: Securing Against AI-Generated Manipulation', Journal of Computational Innovation, 4(1), pp. 1–18.

- Gilbert, C. and Gilbert, M.A. (2024). 'The Role of Artificial Intelligence (AI) in Combatting Deepfakes and Digital Misinformation', *International Research Journal of Advanced Engineering and Science*, 9(4), pp. 170–181.
- Somoray, K., Miller, D.J. and Holmes, M. (2025). 'Human Performance in Deepfake Detection: A Systematic Review', *Human Behavior and Emerging Technologies*, 2025, Article ID 1833228.
- Mohamed, K. and Ozad, B.E. (2025). 'AI-Driven Media Manipulation: Public Awareness, Trust, and the Role of Detection Frameworks in Addressing Deepfake Technologies', *Interdisciplinary Media and Communication Studies*, 2(3), pp. 98–133.
- Atlam, E.-S., Almaliki, M., Elmarhomy, G., Almars, A.M., Elsiddieg, A.M.A. and ElAgamy, R. (2025). 'SLM-DFS: A systematic literature map of deepfake spread on social media', *Alexandria Engineering Journal*, 111, pp. 446–455.
- Seng, L.K., Mamat, N., Abas, H. and Ali, W.N.H.W. (2024). 'AI Integrity Solutions for Deepfake Identification and Prevention', *Open International Journal of Informatics*, 12(1), pp. 35–46.
- Singh, L.H., Charanarur, P. and Chaudhary, N.K. (2025). 'Advancements in detecting Deepfakes: AI algorithms and future prospects - a review', *Discover Internet of Things*, 5(53).
- Zhang, Y., Pang, Z., Huang, S., Wang, C. and Zhou, X. (2025). 'Unmasking AI-Created Visual Content: A Review of Generated Images and Deepfake Detection Technologies', *Preprints.org*. doi: 10.20944/preprints202501.1685.v1.
- Vamsi, V.V.V.N.S., Shet, S.S., Reddy, S.S.M., Rose, S.S., Shetty, S.R., Sathvika, S., Supriya, M.S. and Shankar, S.P. (2022). 'Deepfake detection in digital media forensics', *Global Transitions Proceedings*, 3, pp. 74–79.
- Abir, W.H., Khanam, F.R., Alam, K.N., Hadjouni, M., Elmannai, H., Bourouis, S., Dey, R. and Khan, M.M. (2023). 'Detecting Deepfake Images Using Deep Learning Techniques and Explainable AI Methods', *Intelligent Automation & Soft Computing*, 35(2), pp. 2151–2169.
- Sadiq, S., Aljrees, T. and Ullah, S. (2023). 'Deepfake Detection on Social Media: Leveraging Deep Learning and FastText Embeddings for Identifying Machine-Generated Tweets', *IEEE Access*, 11, pp. 95008–95021.
- Rosca, C.-M., Stancu, A. and Iovanovici, E.M. (2025). 'The New Paradigm of Deepfake Detection at the Text Level', *Applied Sciences*, 15(5), p. 2560.
- Liu, H., Li, J., Gong, C., He, M., Han, K. and Luo, S. (2025). 'When Deepfake Detection Meets Graph Neural Network: a Unified and Lightweight Learning Framework', *arXiv preprint arXiv:2508.05526*.
- He, Y., Cao, Y., Yang, B. and Zhang, Z. (2025). 'Can GPT tell us why these images are synthesized? Empowering Multimodal Large Language Models for Forensics', *ACM Workshop on Information Hiding and Multimedia Security (IH&MMSEC '25)*.
- Khan, A.A., Chen, Y.-L., Hajje, F., Shaikh, A.A., Yang, J., Ku, C.S. and Por, L.Y. (2024). 'Digital forensics for the socio-cyber world (DF-SCW): A novel framework for deepfake multimedia investigation on social media platforms', *Egyptian Informatics Journal*, 27, p. 100502.
- Siegel, D., Kraetzer, C., Seidlitz, S. and Dittmann, J. (2024). 'Media Forensic Considerations of the Usage of Artificial Intelligence Using the Example of DeepFake Detection', *Journal of Imaging*, 10(46).
- Kaur, A., Hoshyar, A.N., Saikrishna, V., Firmin, S. and Xia, F. (2024). 'Deepfake video detection: challenges and opportunities', *Artificial Intelligence Review*, 57(159).
- Khan, N., Nguyen, T., Khalil, I.M. and Bermak, A. (2025). 'Unmasking Synthetic Realities in Generative AI: A Comprehensive Review of Adversarially Robust Deepfake Detection Systems', *arXiv preprint arXiv:2507.21157*.
- Ahmed, S.R., Sonuç, E., Ahmed, M.R. and Duru, A.D. (2022). 'Analysis survey on deepfake detection and recognition with convolutional neural networks', *2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1-7.