

Adversarial Attacks on video Recognition Models



By

Namra Gul
00000328634

Department of Electrical Engineering
School of Electrical Engineering & Computer Science (SEECS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(2024)

Adversarial Attacks on Video Recognition Models



By

Namra Gul

00000328634

A thesis submitted in partial fulfillment of the requirements for the degree of

Master Of Science in

Electrical Engineering (MSEE)

Supervisor

Dr. Arbab Latif

School of Electrical Engineering & Computer Science (SEECS),

National University of Sciences and Technology (NUST),

Islamabad, Pakistan.

(2024)

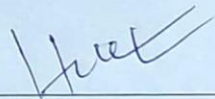
THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Adversarial Attacks on Video Recognition Models" written by Namra Gul, (Registration No 328634), of SEECS has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

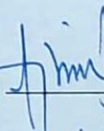
Signature: 

Name of Advisor: Dr. Arbab Latif

Date: 15-Feb-2024

HoD/Associate Dean: 

Date: 25/3/24

Signature (Dean/Principal):  **Dr. Muhammad Ajmal**
Principal,
NUST School of Electrical
& Computer Science
H-12, Islamabad

Date: 25/3/2024

National University of Sciences & Technology
MASTER THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: (Student Name & Reg. #) Namra Gul [328634]

Titled: Adversarial Attacks on Video Recognition Models

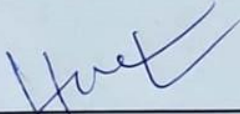
be accepted in partial fulfillment of the requirements for the award of Master of Science (Electrical Engineering) degree.

Examination Committee Members

1. Name: Wajahat Hussain Signature: 
21-Mar-2024 10:14 AM

2. Name: Hafsa Iqbal Signature: 
21-Mar-2024 10:14 AM

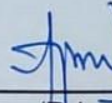
Supervisor's name: Arbab Latif Signature: 
21-Mar-2024 3:52 PM


HoD/Associate Dean

25/3/24
Date

COUNTERSIGNED

25/3/24
Date


Dr. Muhammad Ajmal
Principal,
NUST School of Electrical
& Computer Science

Dean/Principal

Approval

It is certified that the contents and form of the thesis entitled "Adversarial Attacks on Video Recognition Models" submitted by Namra Gul have been found satisfactory for the requirement of the degree.

Advisor : Dr. Arbab Latif

Signature: 

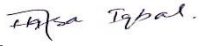
Date: 15-Feb-2024

Committee Member 1:Dr. Wajahat Hussain

Signature: 

14-Feb-2024

Committee Member 2:Dr Hafsa Iqbal

Signature: 

Date: 14-Feb-2024

Dedication

Dedicated to the resilient spirit of countless girls who, due to various circumstances, were unable to pursue their educational aspirations. May this dedication serve as a reminder of the barriers they face and inspire a collective effort towards a world where every girl has the opportunity to learn, grow, and fulfill her potential.

Certificate of Originality

I hereby declare that this submission titled "Adversarial Attacks on Video Recognition Models" is my own work. To the best of my knowledge, it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEecs or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEecs or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation, and linguistics, which has been acknowledged. I also verified the originality of contents through plagiarism software.

Acknowledgments

In the name of ALLAH, the most beneficent and the most merciful who gave me the strength and knowledge to carry out this research and complete the project. My researchwork would not have been completed without the provision and assistance of many people. This is the best time to pay my gratitude to them for their cooperation in this work.

My deepest gratitude and appreciation go to my Project supervisor and mentor **Dr. Arbab Latif** for his constant guidance, motivation, and support during the course of my studies. His valuable suggestions broadened my horizon and made this research work interesting and challenging for me.

I would also like to thank **Dr. Numan Khursheed**, my ex-supervisor, whose initial guidance paved the way for this research.

There are many other people to thank who guided me throughout this research. To the many friends and colleagues who gave feedback, offered suggestions, gave technical help, and provided much needed moral support. I want you to know that I could not have done this without you. Your help has been invaluable, and I am deeply grateful.

My deepest gratitude is extended to my family members, especially to my parents who have been giving me endless care and are always supportive in my education.

Contents

1	Introduction and Motivation	1
1.1	Background:	1
1.2	Adversarial Attacks	3
1.3	Types of Adversarial Attacks	6
1.3.1	White-box Adversarial Attacks	6
1.3.2	Black-box Adversarial Attacks.....	7
1.3.3	Untargeted Attacks	8
1.3.4	Targeted Attacks.....	8
1.4	Motivation	8
1.5	Aims and Objectives	10
1.6	Thesis Layout	11
2	Literature Review.....	12
2.1	Adversarial Attacks on Obfuscated Gradients	13
2.2	Large-Scale Adversarial Attacks and Ensemble Defense	13
2.3	Efficient Video Adversarial Attacks with Keyframe Selection.....	13
2.4	Search-and-Attack Approach for Video Attacks.....	14
2.5	Gradient-Free Decision-Based Black Box Attack with Random Search Optimization.....	15
2.6	PRADA: Black-Box Adversarial Attacks on Neural Ranking Models.....	16
2.7	Exploiting Vulnerabilities in Hashing-Based Video Retrieval.....	17
2.8	Transferable Adversarial Attacks in Object Detection Systems	18
2.9	Reattack: Efficient Black-Box Adversarial Attack with Genetic Algorithms... 19	
2.10	Temporal Translation for Video Adversarial Attacks	21
2.11	Sparked Prior for Motion-Driven Video Adversarial Attacks	22
2.12	Adversarial Attacks on Video Anomaly Detection	23
2.13	Adversarial Attacks in Video Classification using A2F Technique	23
2.14	Bullet-Screen Comment Attacks on Video Recognition Models.....	24
2.15	Efficient Video Anomaly Detection from Weakly Labeled Surveillance Data ..	

.....	25
2.16 Efficient Black-Box Video Adversarial Attacks with GEO- TRAP	27
2.17 Enhancing Robustness in Video Adversarial Attacks using V3A Technique	28
2.18 V-BAD: Black-Box Video Adversarial Attacks and Robustness Evaluation.	29
3_Datasets and Models	32
3.1 Dataset.....	32
3.1.1 Hockey Fight Dataset	32
3.1.2 CrimesScene Dataset	33
3.2 Video Recognition Models.....	35
3.2.1 Convolution 3D Block.....	35
3.2.2 Pseudo 3D Block	37
3.2.3 Quasi 3D Block	38
4_Designing Adversarial Attacks	42
4.1 Adversarial Attacks on Video Recognition Models using Adversarial Patch Technique	42
4.1.1 Methodology.....	43
4.1.2 Consequences and Implications.....	45
4.2 Appending Adversarial Frames to Video Sequences for Adversarial Attacks .	45
4.2.1 Methodology.....	46
4.2.2 Challenges	48
4.3 Adversarial Attacks on Video Recognition Models through Gaussian Noise...	49
4.3.1 Introduction to Gaussian Noise	49
4.3.2 Adversarial Attacks using Gaussian Noise.....	50
4.3.2.1 Methodology.....	50
4.3.3 Consequences and Implications.....	52
4.4 Adversarial Attacks on Video Recognition Models Through Contrast Adjustment	52
4.4.1 Methodology:.....	53
4.4.2 Consequences and Implications.....	55
4.5 Adversarial Attacks on Video Recognition Models using Salt and Pepper Noise	56
4.5.1 Introduction to Salt and Pepper Noise	56
4.5.2 Methodology.....	56
4.5.3 Consequences and Implications.....	59
4.6 Adversarial Attacks on Video Recognition Models using Motion Blur	

Technique	59
4.6.1 Methodology.....	60
4.6.2 Consequences and Implications.....	62
4.7 Adversarial Attacks on Video Recognition Models using Frame Dropping Technique	62
4.7.1 Methodology.....	63
4.7.2 Consequences and Implications.....	64
5 Experiments & Results	65
5.1 Model Performance's Evaluation: Measure of Accuracy	66
5.1.1 Using Accuracy as a Performance Metric	66
5.1.2 Calculation of Accuracy	66
5.1.3 Insights Gained Through Accuracy Assessment in Adversarial Attacks	67
5.2 Performance of the models on CrimesScene dataset before the attack.....	68
5.3 Results of Adversarial Attacks on Crimes Scene Dataset	69
5.3.1 Adversarial Attacks on Video Recognition Models using Adversarial Patch Technique	69
5.3.1.1 Impact of Adversarial Patches in the Bottom right corner	69
5.3.1.2 Impact of Adversarial Patches in the Bottom left corner.....	69
5.3.1.3 Impact of Adversarial Patches in the Top right corner.....	70
5.3.1.4 Impact of Adversarial Patches in the Top left corner.....	70
5.3.2 Adversarial Attacks on Video Recognition Models using Appending Adversarial Frame Technique.....	71
5.3.2.1 Impact of Appending Frames After 30 FPS on Video Recognition Models	72
5.3.2.2 Impact of Appending Frames After 25 FPS on Video Recognition Models	72
5.3.2.3 Impact of Appending Frames After 20 FPS on Video Recognition Models	72
5.3.2.4 Impact of Appending Frames After 15 FPS on Video Recognition Models	72
5.3.3 Adversarial Attacks on Video Recognition Models through Gaussian Noise	74
5.3.3.1 Adversarial Attacks with Low Gaussian Noise Parameters	74
5.3.3.2 Adversarial Attacks Using Gaussian Noise: Low Mu and High Sigma Values	74

CONTENTS

5.3.3.3 Adversarial Attacks Using Gaussian Noise: High Mu and Low Sigma Values	74
5.3.3.4 Adversarial Attacks with High Gaussian Noise Parameters.....	75
5.3.4 Adversarial Attacks on Video Recognition Models Through Contrast Adjustment.....	78
5.3.4.1 Low values of alpha (α) and beta (β).....	78
5.3.4.2 Low values of alpha (α) and High values of beta (β)	78
5.3.4.3 High values of alpha (α) and Low values of beta (β)	79
5.3.4.4 High values of alpha (α) and beta (β)	80
5.3.5 Adversarial Attacks on Video Recognition Models Through Salt & Pepper Noise.....	83
5.3.5.1 Uniform Salt and Pepper Noise Probability in Adversarial Attacks.....	83
5.3.5.2 Varied Salt and Pepper Noise Effects in Adversarial Attacks.....	84
5.3.6 Adversarial Attacks on Video Recognition Models using Motion Blur Technique	86
5.3.6.1 Adversarial Attacks with Horizontal Motion Blur and Varying Kernel Sizes 86	
5.3.6.2 Adversarial Attacks with Vertical Motion Blur and Varying Kernel Sizes .	87
5.3.6.3 Adversarial Attacks with Diagonal (45°) Motion Blur and Varying Kernel Sizes 87	
5.3.6.4 Adversarial Attacks with Diagonal (135°) Motion Blur and Varying Kernel Sizes	88
5.3.6.5 Adversarial Attacks with Cross Motion Blur and Varying Kernel..... Sizes	88
5.3.7 Adversarial Attacks on Video Recognition Models using Frame Dropping Technique	93
5.4 Comparison of Model Performance in Handling Additional Features Related to Learned Patterns	94
5.4.1 Experimental Setup.....	94
5.4.2 Impact on Our Models:	94
5.4.3 Behavior of the Other Model:.....	94
5.4.4 Specifics in the Case of C3D:.....	95
5.4.5 Implications for Our Model:.....	95
5.5 Defenses Against Adversarial Attacks on Video Recognition Models.....	95
5.5.1 Median Filtering	96
5.5.2 Gaussian Blur	96

CONTENTS

5.5.3 Deblurring Filter	97
5.5.4 Experiments & Results	97
6_Conclusion & Future Work.....	100
Bibliography	102

List of Tables

5.1	Performance of Models before the attacks.....	68
5.2	Performance of models after adding the adversarial patches.....	71
5.3	performance of models with appended adversarial frames beyond a certain FPS	73
5.4	performance of the models after adding Gaussian noise with varying parameters.	77
5.5	Performance of the models after changing the contrast by varying different parameters.	82
5.6	performance of the models after adding Salt and Pepper noise with varying parameters.	86
5.7	Performance of the model after the introduction of a motion blur filter in video frames with diverse parameters.	93
5.8	performance of the models after dropping a certain number of frames.....	94
5.9	Performance of our models against these defense strategies	99

List of Figures

1.1: The Chihuahua vs Muffin example.....	5
1.2 White-box attack Scenario	7
1.3 Black-box attack Scenario	7
2.1 Overview of the gradient-based keyframe selection method.....	14
2.2 shows an overview of search-and -attack pipeline.....	15
2.3 Two types of query-based black-box attacks according to adversary’s knowledge (score and decision-based attacks).....	16
2.4 Overall architecture of the PRADA Method.....	17
2.5 The process of querying videos and the creating targeted adversarial videos	18
2.6 The training framework of Unified and Efficient Adversary (UEA).....	19
2.7 MNIST adversarial examples generated by GenAttack.....	20
2.8 Overview of the proposed methodology	21
2.9 Overview of motion-excited sampler for black-box video attack.....	23
2.10 Appending adversarial frame	24
2.11 Overview of black-box adversarial BSC attack method.....	25
2.12 The flow diagram of the proposed anomaly detection approach	26
2.13 Overview of Geo-TRAP	28
2.14 Overview of video-augmentation-based adversarial attack.....	29
2.15 Overview of the proposed V-BAD framework for black-box video attacks	30
3.1 The procedure adapted to annotate frames of video sequence in Normal and Abnormal category. This is activity is repeated for videos of each class	34
3.2 Normal instances from the CrimesScene dataset.....	34

3.3 A simple Conv 3D Block.....	36
3.4 Different variations of the Pseudo-3D Blocks	38
3.5. Different variations of the Q3D block	41
4.1 (a) Toaster-Target b) Crab-Target (c) Toaster-Target (Disguided).	43
4.2 (a) Original frame (b) Perturbed frame after adversarial patch	45
4.3 (a) Original Frame (b) After 3 FPS (c) After 5 FPS	48
4.2 Gaussian Distribution.....	49
4.4 (a) Original Image (b) Perturbed Image after Gaussian Noise	52
4.5 (a) Original Image (b) Perturbed image after contrast adjustment	55
4.2 (a) Before adding salt and pepper noise (b) After adding salt and pepper noise .	56
4.6 (a) Original Image (b) Perturbed image after adding salt & pepper noise.....	59
4.7 (a) Original Image (b) Perturbed image after motion blur.....	62
5.1 Confusion Matrix	67

Abstract

This thesis explores the vulnerability of video recognition models, specifically C3D, P3D, and Q3D, to adversarial attacks using the Crimes Scene dataset. Through rigorous testing involving seven distinct attack strategies, the study investigates the impact on model accuracy, revealing instances where certain attacks consistently lower accuracy and others induce constant effects. Comparative analyses extend to benchmarking the performance of the three models against others within the domain, employing accuracy as a key performance parameter. The findings highlight variations in susceptibility and robustness among the models. Subsequently, it proposes and evaluates defensive strategies aimed at enhancing the resilience of the models against adversarial attacks. This comprehensive examination contributes valuable insights to the field of video recognition model security, offering a nuanced understanding of vulnerabilities, comparative performance, and effective defense mechanisms.

Chapter 1

Introduction and Motivation

1.1 Background:

In the recent years, Deep neural networks (DNNs) have evolved tremendously in terms of architecture design, training techniques, and applications making them a powerful tool in various fields of machine learning and artificial intelligence such as image and video processing, Natural Language Processing (NLP), Speech and Audio processing, health care, autonomous systems, finance, weather forecasting, and many more. The application of deep learning has evolved significantly over the years, driven by advancements in research, algorithms, computing power, and the availability of large datasets.

Deep neural networks (DNNs) have gained immense popularity for their ability to learn complex patterns from data and excel in various tasks. However, their widespread adoption is accompanied by significant vulnerabilities and challenges that can have far-reaching implications for their performance, security, and overall reliability. These vulnerabilities stem from the intricate nature of DNNs, which involve numerous interconnected layers of artificial neurons, making them susceptible to a range of issues. These vulnerabilities are important to be understood and addressed for the safe and responsible deployment of DNNs.

The proliferation of Deep Neural Networks (DNNs) in various domains has undeniably revolutionized the landscape of machine learning and artificial intelligence [34]. These networks, with their deep architectures and intricate training mechanisms, have demonstrated exceptional capabilities in tasks that range from image and speech recognition to complex natural language understanding. Their contributions have permeated almost every facet of modern life, from virtual personal assistants that understand and respond to our voice commands to autonomous vehicles that navigate our streets safely. The transformative potential of DNNs has not only captured the imagination of researchers but has also spurred a wave of innovation in industry and academia alike.

However, with this surge in popularity and integration into real-world applications, DNNs have unveiled a dark underbelly—their susceptibility to adversarial attacks. These attacks exploit the very intricacies that make DNNs so powerful—their sensitivity to subtle patterns and features within data. As DNNs delve deep into high-dimensional decision spaces to make predictions, they become vulnerable to manipulations that may seem imperceptible to human observers but can lead to catastrophic misclassifications.

In real-world scenarios, the impact of adversarial attacks on DNNs can be profound and far-reaching [7]. Consider the use of image recognition systems in autonomous vehicles. These systems play a pivotal role in identifying objects, pedestrians, and other vehicles on the road, contributing to safe navigation and collision avoidance. However, if an adversarial attack can deceive these systems into misclassifying a stop sign as a yield sign or a pedestrian as a lamppost, the consequences can be dire. Such attacks could potentially compromise the safety of autonomous vehicles and put lives at risk.

Similarly, in the realm of healthcare, DNNs are increasingly used for tasks like medical image analysis and disease diagnosis [15]. Adversarial attacks on these systems could lead to misdiagnoses, unnecessary treatments, or even delayed interventions, with grave implications for patient health and well-being.

In the financial sector, where DNNs are deployed for fraud detection and risk assessment, adversarial attacks can result in the evasion of security measures, leading to substantial financial losses. In natural language processing applications, like

sentiment analysis and chatbots, adversarial attacks can manipulate user interactions, spreading misinformation or even causing harm.

These real-world examples underscore the urgency of comprehending and mitigating adversarial vulnerabilities in DNNs. While DNNs have propelled us into an era of unprecedented automation and intelligence, they have also introduced a new dimension of risk that demands careful consideration. To harness the full potential of these powerful tools while ensuring their robustness and reliability, researchers and practitioners must confront the multifaceted challenges posed by adversarial attacks and work collectively to fortify DNNs against this emerging threat. In the following sections of this thesis, we delve deeper into the intricacies of these attacks, their types, and the strategies to enhance the resilience of DNNs in an increasingly adversarial landscape.

1.2 Adversarial Attacks

One critical vulnerability inherent to deep neural networks (DNNs) is their susceptibility to adversarial attacks. Adversarial attacks exploit a fascinating and concerning aspect of DNNs: their remarkable ability to be misled by seemingly imperceptible changes in input data. This vulnerability arises due to the complex, high-dimensional decision boundaries learned by DNNs, making them sensitive to subtle alterations in input features [\[1\]](#).

In the ever-evolving landscape of artificial intelligence and computer vision, the concept of adversarial attacks has garnered significant attention. Adversarial attacks are techniques employed to deceive machine learning models, particularly those used in computer vision, by subtly altering input data. These alterations, often imperceptible to human observers, can lead to incorrect model predictions and, in some cases, serious consequences.

Adversarial examples are carefully crafted data points that are designed to deceive DNNs into making erroneous predictions. These examples are generated by adding perturbations to the original input data in a way that is imperceptible to human observers but significantly alters the DNN's output.[\[3\]](#) The perturbations are typically calculated using optimization techniques to maximize the model's prediction error, causing the DNN to confidently misclassify the input.

The susceptibility of deep neural networks (DNNs) to adversarial attacks is a paradoxical reflection of their own sophistication. These networks, with their intricate architectures, are capable of learning intricate patterns and representations from vast datasets, achieving human-level performance in numerous tasks. However, this very complexity that empowers DNNs also exposes them to manipulation. Adversarial attacks exploit this vulnerability by pinpointing the seams in the DNN's understanding of data, exploiting the chinks in their armor to create seemingly innocuous input modifications that lead to grievous errors in predictions.

Adversarial examples, the ammunition of these attacks, are meticulously engineered to exploit the DNN's blind spots. Crafted with mathematical precision, they introduce subtle perturbations that walk a fine line between human imperceptibility and DNN sensitivity. These perturbations challenge the DNN's fundamental assumption that small input variations should not drastically alter predictions. By shattering this assumption, adversarial examples reveal the brittleness of DNNs, demonstrating that even state-of-the-art models can be led astray by these carefully constructed deceptions [\[5\]](#).

In essence, adversarial attacks lay bare the paradox of DNNs: their astonishing capabilities coexist with their vulnerability to minute alterations in input data. The ramifications of this vulnerability extend across domains where DNNs are employed, from autonomous vehicles making life-or-death decisions on the road to medical diagnoses and financial transactions that impact individuals and societies. As we delve deeper into the realm of adversarial attacks, we aim to uncover not only their mechanisms but also strategies to fortify DNNs against this intriguing and unsettling threat.

Example:

Figure 1.1 shows how a simple modification in an image can lead to an adversarial attack.



Fig 1.1: The Chihuahua vs Muffin example

In the context of an adversarial attack, the "Chihuahua or muffin" example can be used to highlight the challenges faced in developing robust machine learning models. For instance, an attacker could create subtle modifications to an image of a Chihuahua that, when processed by a machine learning model trained to classify dogs, could cause it to misclassify the image as a muffin. This manipulation could involve perturbing pixel values or adding imperceptible noise to the image.

The "Chihuahua or muffin" example serves as a reminder that even seemingly straightforward tasks, such as distinguishing between dogs and muffins, can be challenging for machine learning models. Adversarial attacks highlight the need for developing more robust and resilient models that can withstand intentional manipulations and maintain accurate predictions.

The goal of such attacks is to exploit vulnerabilities in the model's decision-making process and expose its lack of robustness. By presenting a seemingly innocuous image that the model fails to classify correctly, the attacker can demonstrate the model's susceptibility to manipulation and potentially use this knowledge maliciously.

Another example is the use of adversarial attacks in security systems where it could be used to manipulate facial recognition systems, biometric authentication, or intrusion detection. An attacker could alter an image of an authorized individual to gain unauthorized access or evade detection.

1.3 Types of Adversarial Attacks

There are two main types of adversarial attacks based on the attacker's knowledge.

1.3.1 White-box Adversarial Attacks

White-box adversarial attacks are a category of attacks on machine learning models, particularly deep neural networks, where the attacker has complete access to the target model's architecture, parameters, training data, and decision-making process. The attackers study the model's architecture, calculate gradients to understand input-output relationships, optimize perturbations to create misleading examples, and test these modifications to see if they trick the model into making incorrect predictions. The primary goal of a white-box adversarial attack is to generate carefully crafted perturbations or modifications to the input data (such as images or text) to deceive the model into making incorrect predictions [29]. Figure 1.2 shows a white-box attack scenario.

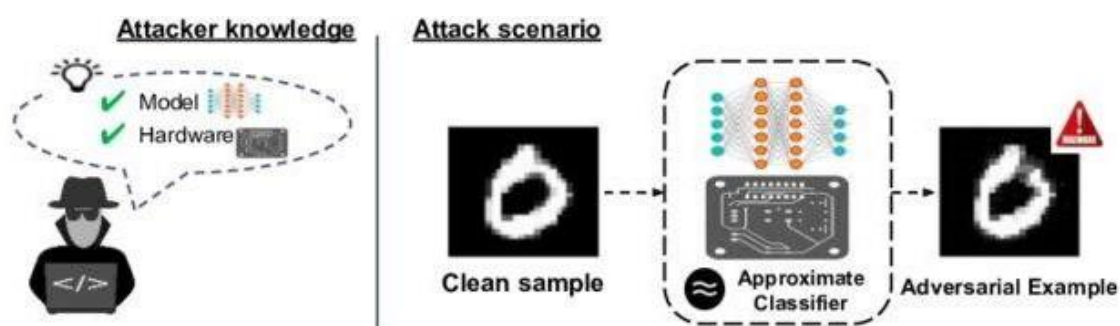


Figure 1.2 White-box attack Scenario

1.3.2 Black-box Adversarial Attacks

Black-box adversarial attacks are a type of attack on machine learning models where the attacker has limited or no access to the inner workings of the target model. Black-box attacks assume that the attacker can only observe the model's input-output behavior without knowing its internal details.

In black-box attacks, attackers create a surrogate model imitating the target, use it to query the target for responses, exploit transferability, optimize input changes to deceive the target, and assess success by testing modified inputs for surprising model predictions. The objective of a black-box adversarial attack is to craft inputs (such as images, text, or other data) that can cause the target model to make incorrect predictions or produce unexpected outputs. Since the attacker doesn't know the exact details of the model, they rely on various techniques and strategies to generate these adversarial examples [28]. Figure 1.3 shows a black-box attack scenario.

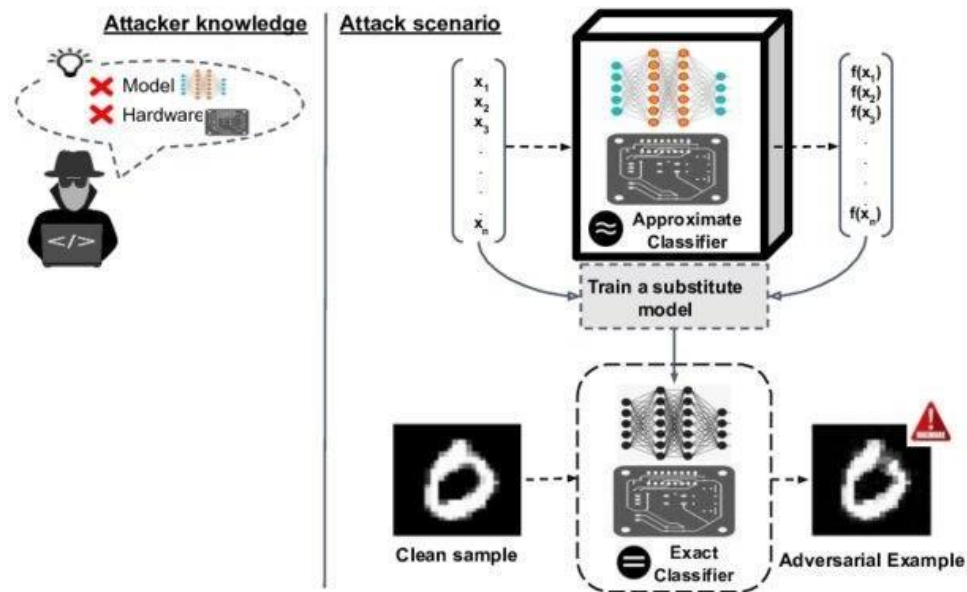


Figure 1.3 Black-box attack Scenario

There are further two types of adversarial attacks whose primary distinction arises from the intended objective of the attacker.

1.3.3 Untargeted Attacks

Untargeted attacks are concerned with manipulating input data in a way that causes a classifier, often a Deep Neural Network (DNN) classifier, to make an incorrect prediction about the object in the image. The goal here is to introduce subtle changes that result in any form of misclassification. The focus is on exploiting the model's vulnerabilities and pushing it into making a mistake, regardless of which incorrect category it assigns to the object [26]. In untargeted attacks, the intent is to demonstrate the model's susceptibility to adversarial perturbations and showcase its potential for making errors in its decision-making process.

1.3.4 Targeted Attacks

Targeted adversarial attacks involve the deliberate manipulation of input data, with the specific goal of causing a machine learning model, like a Deep Neural Network (DNN) classifier, to predict a predetermined incorrect output class. Unlike untargeted attacks, where the aim is to induce any form of misclassification, targeted attacks focus on steering the model's prediction towards a particular, predefined misclassification. This requires crafting precise adversarial examples that guide the model's decision boundaries in the desired direction. [27] The success of a targeted attack is measured by the ability to make the model confidently predict the specified incorrect class, highlighting the model's susceptibility to manipulation and the potential consequences of such attacks in practical applications.

1.4 Motivation

While adversarial attacks on images have been extensively explored, the domain of video adversarial attacks remains relatively unexplored. Adversarial attacks on video recognition models still pose a significant and challenging problem in the field of computer vision. Video recognition models, which are designed to understand and classify the content of videos, are vulnerable to similar types of attacks that affect image recognition models [4].

Attacks on videos can confuse models by changing things over time, altering frames, appearances, or motion patterns. Even the sequence and timing of actions can be messed

up, leading to wrong predictions. These attacks can target individual frames (spatial attacks) or manipulate the sequence of frames (temporal attacks), revealing vulnerabilities in the models' decision-making. These tricks might work in one part of the video and still fool the model in other parts. This is risky, like tricking security cameras or causing issues with self-driving cars. Smart solutions are being developed to tackle these challenges, making models understand motion and time better, to ensure they stay accurate and reliable in real-life situations.

However, the dynamic and temporal nature of videos introduces additional and unique complexities to the attack scenarios compared to image-based attacks. Videos are essentially a series of pictures shown one after another quickly. Since videos are made up of frames, what happens in one frame is often related to what happens in the next frame. This connection between neighboring frames is like a thread that holds the whole video together. This close relationship between neighboring frames is important when dealing with adversarial attacks because any changes we make to one frame might affect the way things look and move in the frames that come after it. This complexity makes crafting effective adversarial attacks on videos quite challenging. Video recognition models are vital in diverse fields, from surveillance to entertainment, yet their vulnerability to adversarial attacks raises concerns that must be addressed. Performing adversarial attacks on videos presents a number of challenges that require elaborate solutions.[\[19\]](#) These challenges span various dimensions, including temporal consistency, computational intensity, dynamic motion understanding, temporal relationship comprehension, real-time constraints, evaluation metric robustness, contextual generalization, model transferability, variability accommodation, privacy concerns, perceptibility to human observers, and computational demands.

Adversarial perturbations must intricately maintain temporal consistency across multiple frames to avoid disrupting the natural video flow, demanding a deeper understanding of dynamic motion and context. Optimizing perturbations in both spatial and temporal dimensions is computationally intensive, requiring specialized algorithms and substantial computational resources, all while considering real-time constraints in applications like surveillance. Furthermore, crafting effective adversarial examples across different time points amidst evolving video content poses a challenge. Defining robust evaluation metrics for quantifying attack effectiveness and impact on recognition

models is complex, particularly for diverse and complex video content. The ability of adversarial attacks to generalize across varying video contexts, scenes, lighting conditions, and camera angles is inherently intricate. Privacy concerns emerge as video manipulation raises ethical considerations, especially in personal videos or surveillance footage. The perceptibility of these attacks to human observers due to noticeable changes in object appearances, motion, and timing sets them apart from attacks on static images. Altogether, the high computational cost and multifaceted nature of video-based adversarial attacks underscore the need for sophisticated techniques and comprehensive understanding to navigate these challenges effectively.

The overall research work can be summed-up as follows:

1. To investigate the susceptibility of video recognition models to adversarial attacks, focusing on dynamic content and temporal relationships.
2. To analyze the intricacies of crafting effective perturbations across spatial and temporal dimensions, considering the unique challenges posed by video data.
3. Assess the impact of adversarial attacks on real-world applications like safety and surveillance systems, highlighting potential vulnerabilities.
4. Propose and develop novel techniques to enhance the robustness of video recognition models against adversarial attacks, ensuring reliable performance in dynamic visual contexts.

1.5 Aims and Objectives

The study systematically investigates adversarial threats on video recognition models, uncovering vulnerabilities and emphasizing the need for enhanced robustness in dynamic scenarios, such as surveillance and autonomous vehicles.

Contributions of our work are:

- The research evaluates a diverse set of adversarial techniques, including the Adversarial Patch Technique, noise injection, frame manipulation, and motion

blur. This comprehensive assessment simulates real-world challenges, providing valuable insights into model-specific vulnerabilities.

- The findings reveal distinct responses of Convolution 3D, Pseudo 3D, and Quasi 3D Models to various adversarial methodologies. This in-depth analysis highlights model-specific weaknesses, contributing to a nuanced understanding of video recognition model behavior under adversarial threats.
- We have introduced novel defense strategies, including median filtering, Gaussian blur, and deblurring filters, which proposes effective countermeasures against adversarial perturbations. Notably, the study identifies median filtering as a particularly robust defense mechanism, enhancing model resilience.
- The study extends beyond theoretical insights, providing practical implications for deploying video recognition systems in security and law enforcement. The research emphasizes the significance of understanding model limitations and implementing defense strategies to ensure reliable and secure video recognition in real-world applications.

1.6 Thesis Layout

The rest of the thesis is organized as follows:

This introductory chapter explains our research area, problem statement, and the scope of this research project. It also includes a brief explanation of the different types of adversarial attacks. Chapter 2 provides a review of the recent literature in the field of adversarial attacks on images and videos. Chapter 3 includes a detailed explanation of the dataset and models that we have used in this work. This is followed by Chapter 4, which includes the methodology, and state-of-the-art performances in the literature. Chapter 5 deals with the experiments and results analysis based on the model's prediction and presenting making the DNNs resilient against those attacks. In the end, Chapter 6 discusses the conclusions of this research work and provides some suggestions for the future work.

Chapter 2

Literature Review

The growing prevalence of video recognition models across diverse applications has fueled significant advancements in computer vision. However, as these models become increasingly integrated into critical systems such as surveillance, autonomous vehicles, and video analysis, concerns surrounding their vulnerability to adversarial attacks have gained prominence. Adversarial attacks, well-documented in the context of image-based models, pose unique challenges when extended to video data due to the temporal and spatial intricacies inherent in video sequences. In this literature review, we delve into the evolving landscape of adversarial attacks specifically tailored to video recognition models. By examining the methodologies, techniques, and implications of such attacks, this review aims to provide a comprehensive understanding of the current state-of-the-art, identify research gaps, and shed light on potential strategies for fortifying video recognition models against adversarial threats.

Furthermore, this review will explore the various categories of adversarial attacks on video recognition models, including spatial and temporal perturbations, and their corresponding impact on model performance. Spatial attacks focus on subtly altering individual frames to deceive the model's perception, while temporal attacks exploit the sequential nature of videos to disrupt the model's understanding of motion and scene progression. By analyzing the strengths and limitations of these attack types, we aim to uncover insights into the unique challenges posed by video data and how they differ from image-based adversarial attacks. Additionally, this review will delve into the transferability of adversarial attacks across different video recognition architectures and explore the effectiveness of defenses proposed to mitigate the impact of such attacks. By synthesizing the existing literature on this evolving topic, we endeavor to provide a comprehensive overview that informs future research

directions and facilitates the development of robust and secure video recognition models in the face of adversarial manipulation.

2.1 Adversarial Attacks on Obfuscated Gradients

Athalye, Carlini, and Wagner [1], investigate the effectiveness of various defense mechanisms designed to protect deep learning models from adversarial attacks. They focus on a method known as "obfuscated gradients," which aims to mislead attackers by making the gradients of the model difficult to interpret. The authors demonstrate that despite the perceived security of these obfuscated gradients, skilled attackers can still find ways to generate adversarial examples that can successfully bypass the defenses. Their findings highlight the importance of thoroughly evaluating the robustness of defense strategies and suggest that relying solely on obfuscated gradients may not provide a strong defense against determined adversarial attacks. The paper underscores the ongoing challenge of developing defenses that can effectively withstand sophisticated adversarial manipulation in deep learning models.

2.2 Large-Scale Adversarial Attacks and Ensemble Defense

In another work by Kurakin, Goodfellow, and Bengio [2], they examine the susceptibility of deep neural networks to adversarial attacks in large-scale scenarios. The study introduces the Basic Iterative Method, an extension of FGSM, to generate subtle perturbations causing significant misclassifications. The authors emphasize the potential impact of these attacks on security and privacy in real-world applications. The paper also proposes "Ensemble Adversarial Training," demonstrating its effectiveness in enhancing model resilience against adversarial examples.

2.3 Efficient Video Adversarial Attacks with Keyframe Selection

The work by Xu, Liu, Yin, Hu, and Ding [3] introduces an innovative approach to generating subtle adversarial perturbations in videos. By strategically selecting keyframes based on gradient analysis, the method efficiently manipulates specific frames to mislead deep neural network models. The proposed sparse adversarial attacks achieve comparable performance to dense attacks while utilizing fewer perturbations, highlighting the vulnerability of video recognition models to subtle adversarial

manipulation. This work contributes to advancing our understanding of adversarial attacks in the context of video data and underscores the importance of robustness in video recognition models. Figure 2.1 provides an overview of the gradient-based keyframe selection method.

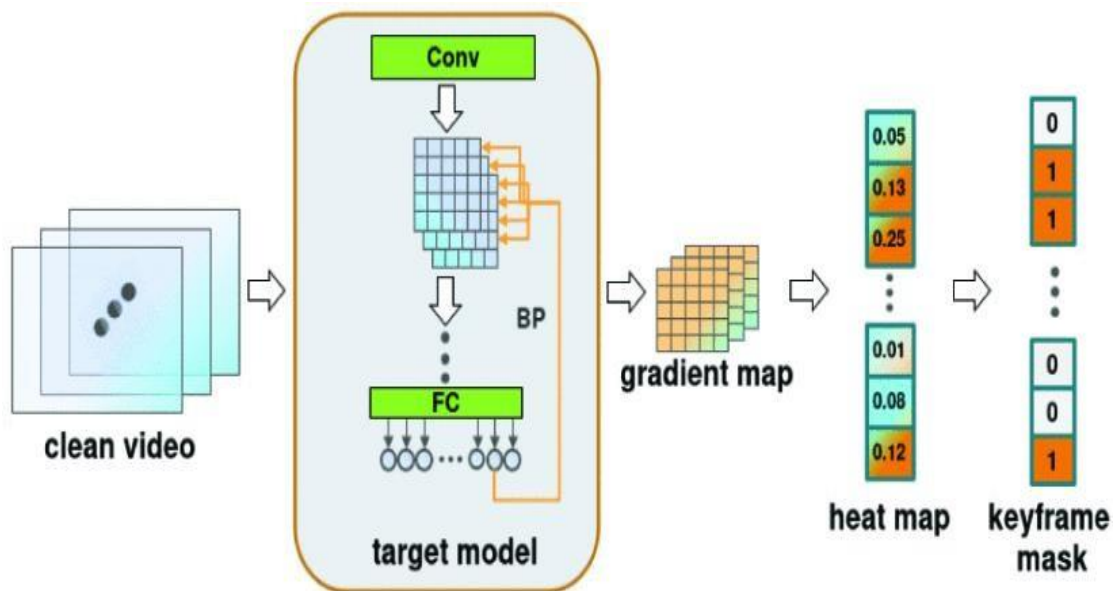


Figure 2.1 Overview of the gradient-based keyframe selection method

2.4 Search-and-Attack Approach for Video Attacks

The research paper by Heo, Ko, Lee [4] explores a method for inserting strategically timed disruptions into videos, known as adversarial perturbations. These disruptions aim to deceive video analysis systems while minimizing their visibility to human viewers. Referred to as "search-and-attack," this technique identifies key moments to insert perturbations, potentially causing misclassification by video analysis algorithms thus reducing the computational cost of the attack. The paper highlights the vulnerability of video analysis systems to these temporally sparse perturbations and discusses their potential implications on security and surveillance. In essence, the paper contributes to the field of adversarial machine learning by addressing challenges specific to videos and proposing strategies to enhance the resilience of video analysis systems. Figure 2.2 shows an overview of search-and -attack pipeline.

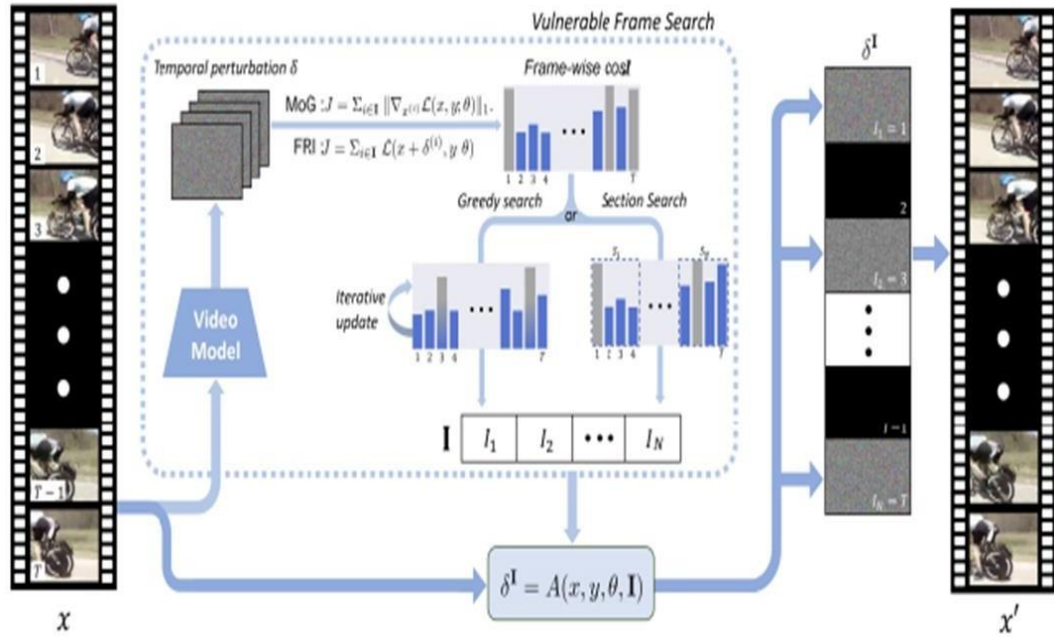


Figure 2.2 shows an overview of search-and-attack pipeline

2.5 Gradient-Free Decision-Based Black Box Attack with Random Search Optimization

In another work by Kim, Yu, and Ro [5], the authors explore a new method for attacking machine learning models without knowledge of their internal parameters by presenting a Gradient-free decision-based and boundary-free black box attack using random search optimization. It employs a coarse-to-fine random search technique to create adversarial perturbations that deceive the model's predictions. The approach focuses on a decision-based setting, utilizing only output labels, and gradually refines perturbations in a targeted manner. This strategy efficiently discovers robust adversarial examples with fewer queries to the target model, demonstrating its effectiveness through experiments. Figure 2.3 shows two types of query-based black-box attacks according to adversary's knowledge (score and decision based attacks).

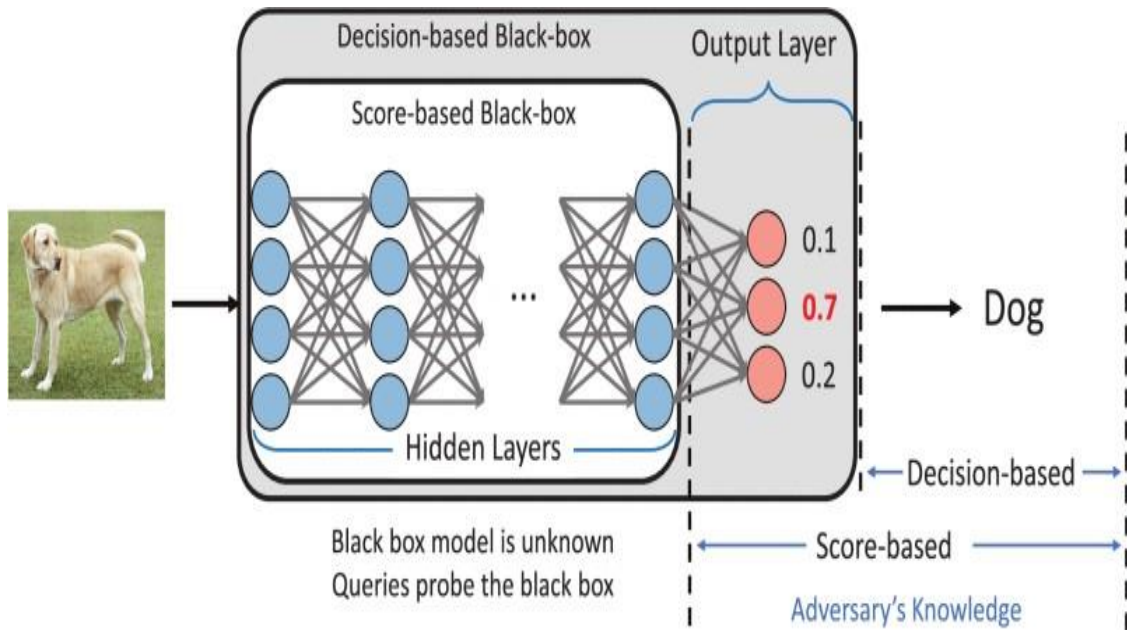


Figure 2.3 Two types of query-based black-box attacks according to adversary's knowledge (score and decision-based attacks)

2.6 PRADA: Black-Box Adversarial Attacks on Neural Ranking Models

In the work by Wu, Zhang and Guo [6], they introduced a method called "PRADA: Practical Black-box Adversarial Attacks against Neural Ranking Models" PRADA for conducting effective black-box adversarial attacks on neural ranking models. The approach generates adversarial queries to manipulate model rankings without needing access to internal parameters. PRADA utilizes gradient-based optimization for perturbing query terms, ensuring practicality by requiring only query-level access. The paper demonstrates the attack's effectiveness through experiments on various neural ranking models, emphasizing its real-world relevance and the need to enhance model robustness against such attacks in critical domains like search engines and recommendation systems. Figure 2.4 shows overall architecture of the PRADA Method.

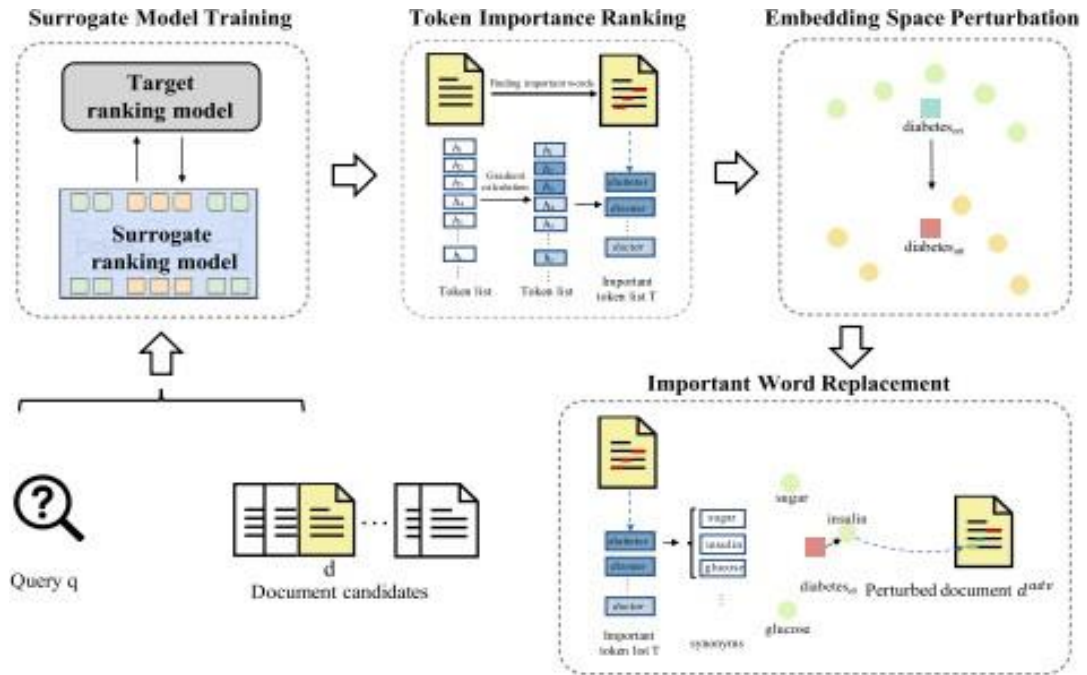


Figure 2.4 Overall architecture of the PRADA Method

2.7 Exploiting Vulnerabilities in Hashing-Based Video Retrieval

In the work by Hu, Huang, Shi [71], the primary objective is to explore and exploit the vulnerabilities of a hashing-based video retrieval system. This type of system employs a technique called hashing to efficiently store and retrieve videos based on their content. Hashing involves converting complex data like videos into compact binary codes, allowing for quick and accurate similarity comparisons. The researchers focus on a specific strategy to undermine the accuracy of this video retrieval system. They choose to manipulate the last 8 frames of a video. These frames are crucial in capturing the conclusion or final moments of the video, which often hold important information about its content. By perturbing or altering the last 8 frames of a video in subtle and carefully calculated ways, the researchers aim to trick the hashing-based system into producing incorrect binary codes for these videos. As a result, when the system attempts to retrieve videos based on similarity, it might yield inaccurate results, associating videos with incorrect or unrelated content. This process effectively demonstrates a form of adversarial attack, where small changes to the input data can lead to significant misclassifications or errors in the system's output. Such attacks highlight potential vulnerabilities that hashing-based video retrieval systems might have against targeted

perturbations. The research underscores the need for developing robust defenses against these types of attacks to ensure the reliability and accuracy of video retrieval systems. Enhancing the resilience of such systems against adversarial perturbations becomes crucial to maintaining their effectiveness in real-world applications where accurate video retrieval is essential, such as video surveillance, content recommendation, or search engines. Figure 2.5 shows the process of querying videos and the creating targeted adversarial videos.

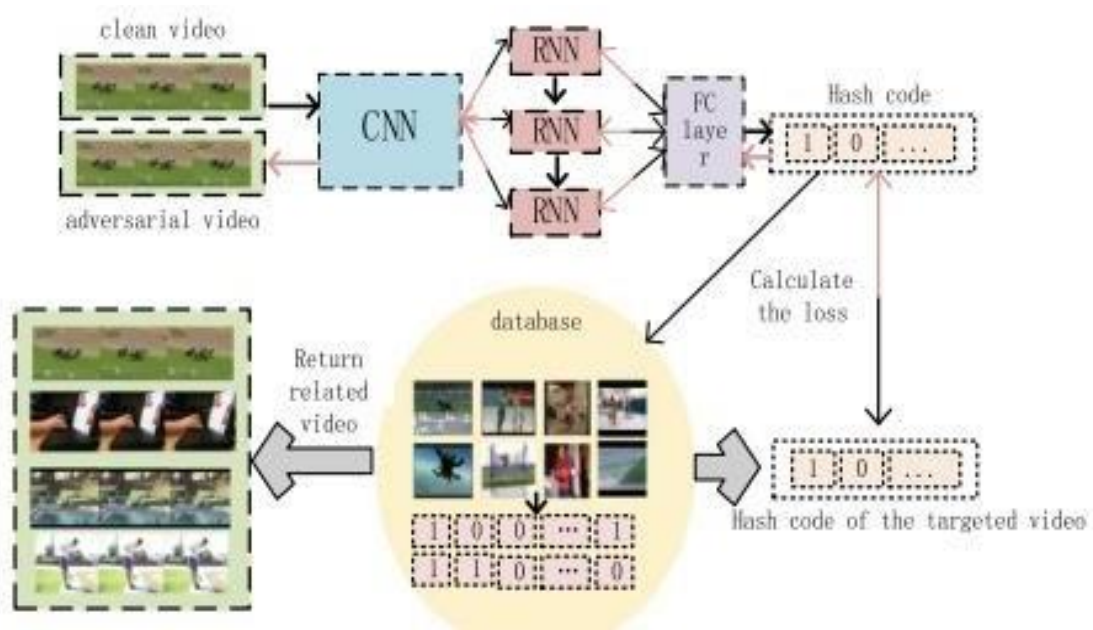


Figure 2.5 The process of querying videos and the creating targeted adversarial videos

2.8 Transferable Adversarial Attacks in Object Detection Systems

In another work by Wei, Liang, Chen [8], the researchers focus on the creation of transferable adversarial attacks targeting image and video object detection systems. These attacks involve generating perturbed images and videos that can fool object detection models, even when the models were not originally trained to recognize such adversarial examples. The research explores the transferability of adversarial attacks across different object detection models, architectures, and datasets. The goal is to demonstrate that adversarial perturbations crafted for one model can effectively deceive other models, highlighting a potential vulnerability in the generalization capability of

object detection systems. The paper likely includes experimental results that showcase the success of transferable adversarial attacks in causing multiple object detection models to misclassify or fail to detect objects. These findings emphasize the importance of developing robust defenses against adversarial attacks in object detection tasks, as well as the need to enhance the models' ability to handle diverse and unexpected inputs. In essence, the research contributes to the field of adversarial machine learning by revealing the cross-model vulnerability of object detection systems and underscoring the significance of creating more resilient models to maintain accurate object detection performance in various real-world scenarios. Figure 2.6 shows the training framework of Unified and Efficient Adversary (UEA).

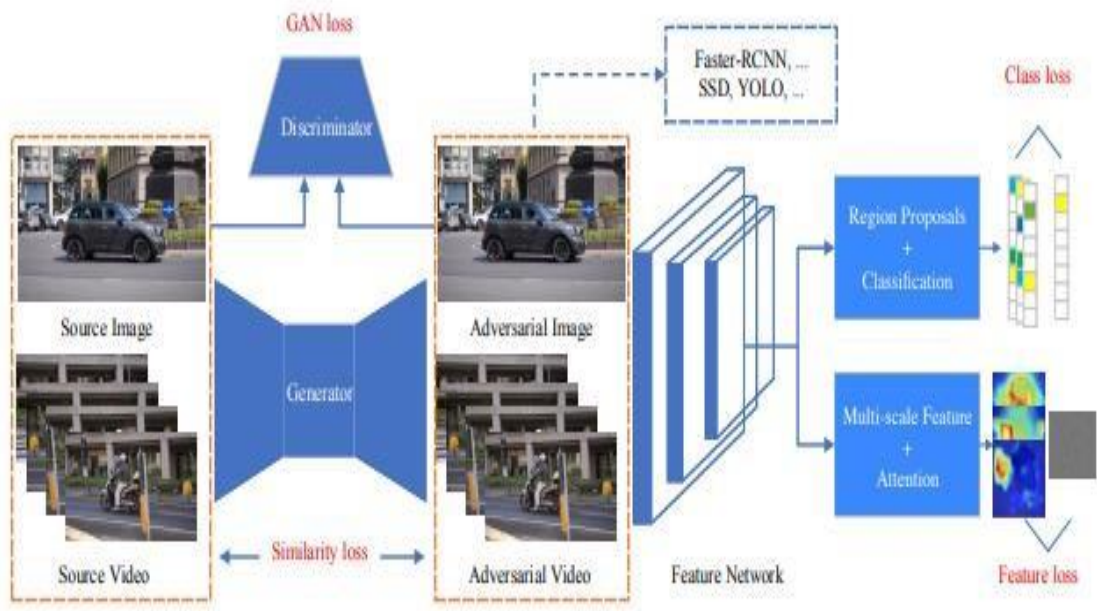


Figure 2.6 The training framework of Unified and Efficient Adversary (UEA)

2.9 Reattack: Efficient Black-Box Adversarial Attack with Genetic Algorithms

In the work by Alzantot, Sharma, and Chakraborty [9], the authors introduced a pioneering method for generating adversarial examples in the challenging black-box setting. The innovation lies in utilizing genetic algorithms, a gradient-free optimization technique, to efficiently create these adversarial examples without requiring access to the target model's internal details. This method iteratively refines a population of feasible adversarial solutions to achieve success. In various experiments, GenAttack

demonstrates remarkable query efficiency compared to existing black-box attack methods by performing practical experiments on diverse datasets including MNIST, CIFAR-10, and ImageNet, drastically reducing the number of queries required for adversarial example generation. This approach performs well even on high-dimensional datasets like ImageNet and remains effective against defenses that manipulate gradients. In essence, GenAttack presents a practical and efficient solution to bolster the security and robustness of neural networks against adversarial attacks.

Figure 2.7 shows MNIST adversarial examples generated by GenAttack where Row label is the True label and Column label is the target label.

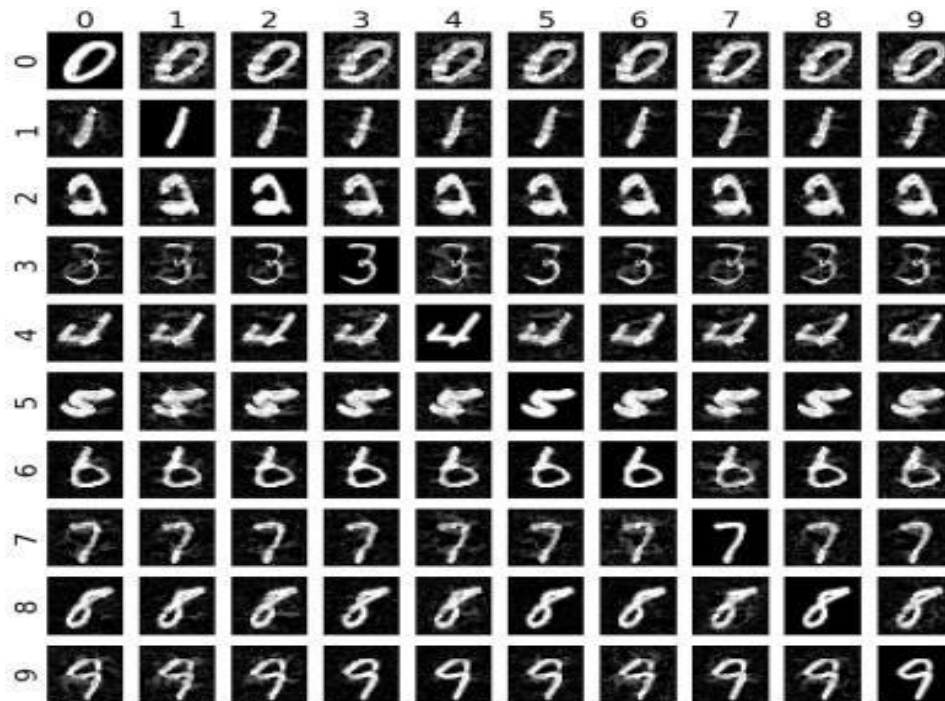


Figure 2.7 MNIST adversarial examples generated by GenAttack.

By leveraging genetic algorithms and avoiding the need for gradients, GenAttack offers a practical and efficient solution for crafting adversarial examples in real-world scenarios where model internals are not accessible. Its ability to significantly reduce query requirements and effectively target high-dimensional models highlights its potential impact on enhancing the security and robustness of deep neural networks.

2.10 Temporal Translation for Video Adversarial Attacks

In the work by Wei, Chen, and Wu [10], the authors introduce a pioneering exploration of transfer-based attacks on videos, a relatively uncharted research area. It presents a temporal translation attack method, a novel approach aimed at enhancing the transferability of adversarial examples for video recognition models operating in a black-box scenario. The authors conduct an insightful analysis of discriminative temporal patterns across diverse video recognition models. This analysis illuminates the complexity of achieving cross-model transferability and serves as a foundation for their innovative solution. Drawing inspiration from spatial translations in image attacks, the paper proposes a temporal translation approach. By optimizing adversarial examples over temporally translated video clips, this method diminishes sensitivity to specific temporal patterns. This leads to the creation of more transferable adversarial examples. The proposed method undergoes rigorous empirical evaluation, involving six video recognition models and datasets (Kinetics-400 and UCF-101). The experimental outcomes validate the effectiveness of the temporal translation approach, demonstrating substantial enhancements in the transferability of video adversarial examples.

Figure 2.8 shows an overview of the proposed methodology.

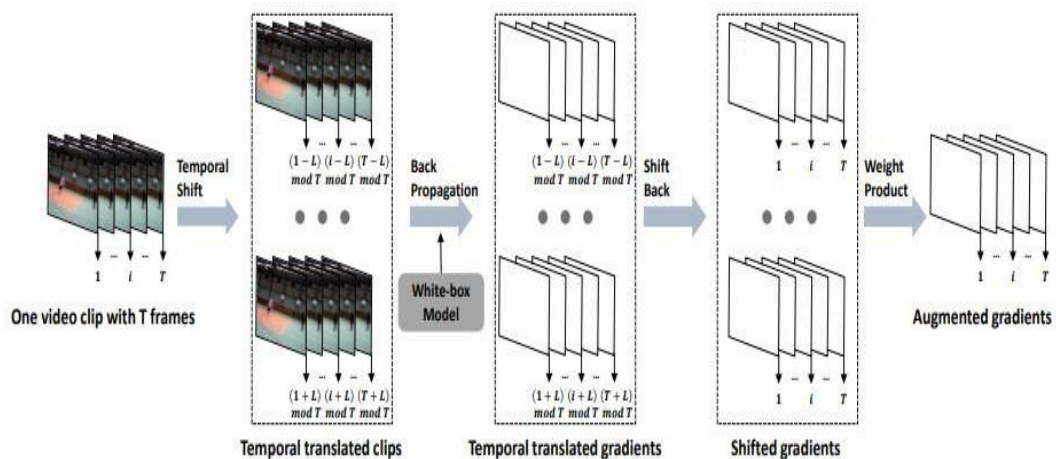


Figure 2.8 Overview of the proposed methodology

2.11 Sparked Prior for Motion-Driven Video Adversarial Attacks

In the study by Zhang, Zu, and Yang [12], the authors introduce a "Sparked Prior Incorporation" technique that enhances the generation of adversarial perturbations in videos. This innovative approach incorporates a sparked prior to capture motion information, contributing to more successful attacks by aligning perturbations with inherent motion patterns within videos. The resulting "Motion-Driven Perturbations" strategy creates convincing and evasive adversarial examples that exploit motion patterns while maintaining "Visual Imperceptibility" to human observers. The method prioritizes perturbations that manipulate video recognition models effectively without being easily detected. Extensive "Empirical Validation" through experiments demonstrates the efficacy of motion-driven perturbations and the sparked prior in creating potent adversarial attacks against video recognition systems. In summary, this approach introduces a novel way to leverage motion information for generating robust and visually imperceptible adversarial examples, thereby enhancing their effectiveness in video-based attacks. By aligning perturbations with motion patterns, the technique achieves a balance between visual imperceptibility and attack potency. The empirical validations highlight the efficacy of this method in creating deceptive adversarial attacks against video recognition models.

Figure 2.9 shows an overview of motion-excited sampler for black-box video attack.

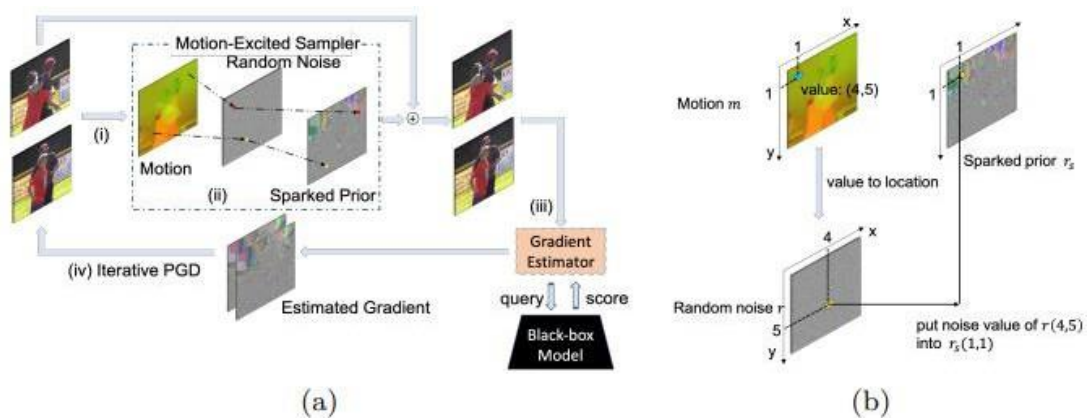


Figure 2.9 Overview of motion-excited sampler for black-box video attack

2.12 Adversarial Attacks on Video Anomaly Detection

In this work, the authors Mumcu et.al [13] presents a comprehensive exploration of adversarial attacks targeted at video anomaly detection systems. These attacks involve the manipulation of video data to deceive anomaly detection models. The study aims to exploit model vulnerabilities by subtly altering input videos to evade anomaly detection. It analyzes the susceptibility of video anomaly detection models to adversarial perturbations, investigating their accuracy in detecting anomalies and highlighting weaknesses. Various techniques are explored for crafting adversarial perturbations, involving pixel manipulation and subtle alterations to video frames. Empirical experiments rigorously evaluate the effectiveness of proposed attacks, assessing their impact on model performance and robustness. The paper offers practical insights into the security of video anomaly detection, underscoring the risks of adversarial attacks and advocating for the development of more resilient detection models.

2.13 Adversarial Attacks in Video Classification using A2F Technique

In the work by Chen, Xie, Pang [14], the authors delve into the realm of generating adversarial examples for video classification. It introduces a novel approach that leverages the semantic and perception spaces to manipulate video content, thereby evading detection in video anomaly detection systems. This study introduces the innovative Appending Adversarial Frames (A2F) approach, which involves strategically replacing consecutive frames in a video with dummy content, followed by the addition of adversarial perturbations solely to these modified frames as shown in Figure 2.11 adversarial examples are generated for video classification by replacing the ending part of the input clip with a few input frames. This two-step process effectively pushes the video closer to the classification border, enhancing the attack's success rate and minimizing perceptibility. The paper discusses various application scenarios for A2F as well, with a particular emphasis on black-box attacks. This approach is demonstrated to achieve high success rates across six state-of-the-art video

classification networks. Moreover, the approach exhibits remarkable transferability, enabling successful attacks across diverse videos and models, highlighting its potential as a universal adversarial attack method. In conclusion, this research establishes a conceptual framework by projecting videos into semantic and perception spaces. It underscores the distinct characteristics of video-based adversarial attacks compared to image-based attacks, emphasizing the significance of considering the inherent structure of videos.

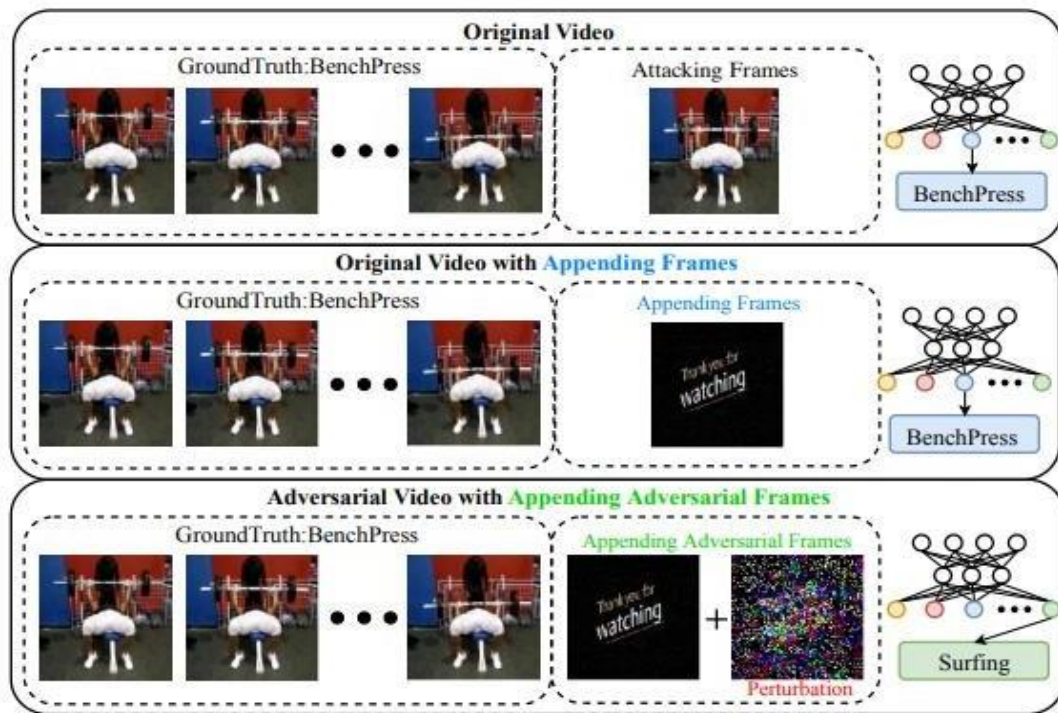


Figure 2.10 Appending adversarial frame

2.14 Bullet-Screen Comment Attacks on Video Recognition

Models

In the work by Chen, Wei, Wu [15], the authors investigate the novel concept of adversarial attacks on video recognition models using bullet-screen comments (BSCs). In this method, the authors focus on patch-based attacks for videos in the black-box setting. This method works by formulating the attack process as a Reinforcement Learning (RL) problem, the authors enable an efficient search for optimal BSC positions and transparencies. The agent, driven by rewards based on fooling rate and

BSC overlap, adapts its selection strategy to achieve effective attacks. Extensive experiments on prevalent video recognition models and benchmark datasets (UCF-101, HMDB-51) demonstrate the BSC attack's efficacy. BSCs, resembling meaningful annotations, are attached to videos, making them less perceptible and more authentic. The method achieves high fooling rates while occluding only a small fraction of the video content. In conclusion, the proposed approach significantly advances the understanding of adversarial attacks on video recognition models, providing a valuable contribution to the field's ongoing research.

The proposed methodology is shown in Figure 2.11.

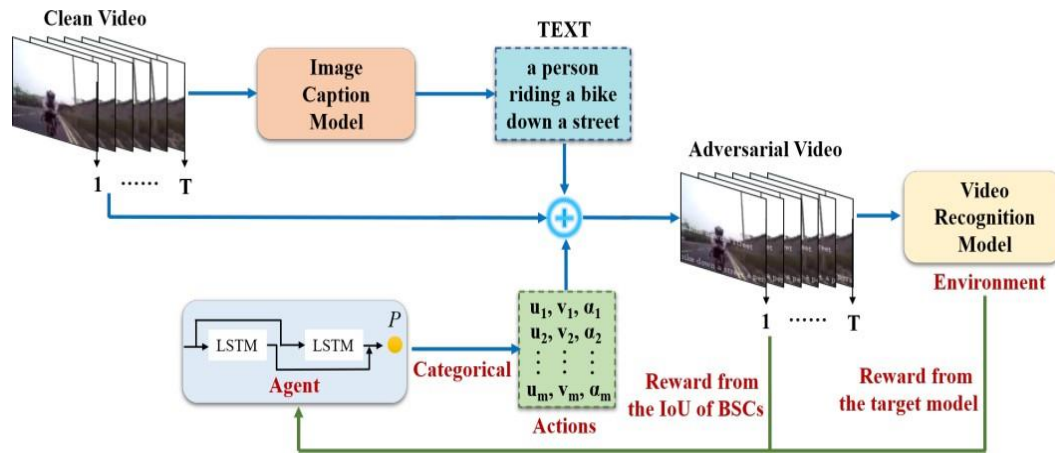


Figure 2.11 Overview of black-box adversarial BSC attack method

2.15 Efficient Video Anomaly Detection from Weakly Labeled Surveillance Data

In the work by Sultani et. al [16], the authors focused on video anomaly detection using surveillance videos. The authors propose a method to learn anomalies by utilizing both normal and anomalous videos. Instead of annotating anomalous segments within training videos, which is time-consuming, the authors suggest leveraging weakly labeled training videos where the labels are assigned at the video level rather than the clip level. The approach employs a deep multiple instances ranking framework for learning anomalies and automatically generating anomaly scores for video segments. Sparsity and temporal smoothness constraints are incorporated into the ranking loss function to improve anomaly localization during training. The authors introduce a new,

extensive dataset comprising 1900 real-world surveillance videos, totaling 128 hours of footage. This dataset features 13 distinct realistic anomalies such as fighting, road accidents, burglary, and robbery, alongside normal activities. The dataset can be utilized for two primary tasks: general anomaly detection and recognizing each of the 13 anomalous activities. The paper's experimental results demonstrate that their multiple instance learning (MIL) method for anomaly detection outperforms existing approaches. Additionally, the dataset serves as a challenging benchmark for activity recognition due to the complexity and intra-class variations of activities. Baseline methods such as C3D and TCNN are tested on recognizing the 13 different anomalous activities. The authors also address the increasing use of surveillance cameras in public spaces for safety purposes and highlight the need for efficient video anomaly detection. The authors emphasize that practical anomaly detection systems should not rely heavily on prior event information and should ideally require minimal supervision. Sparse-coding-based approaches are mentioned as promising methods for anomaly detection, but they can struggle with changing environments and false alarms.

The flow diagram of the proposed anomaly detection approach is shown in Figure 2.12.

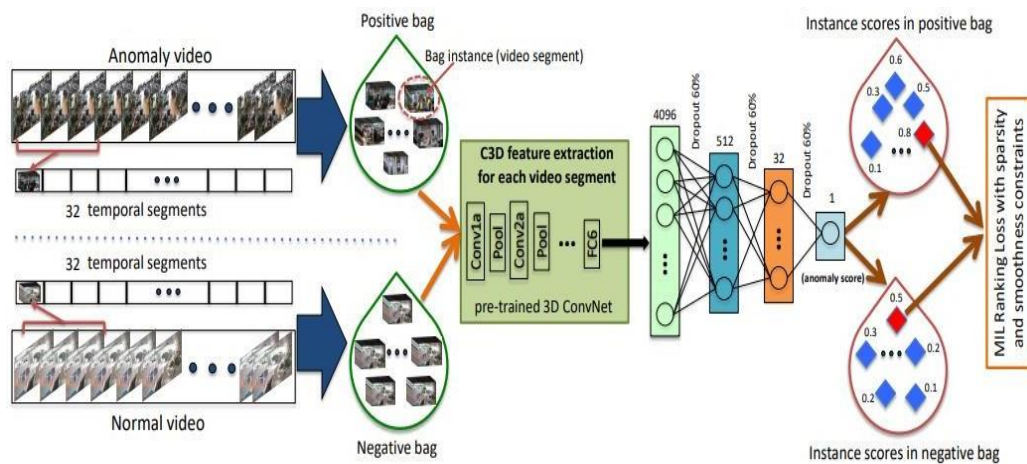


Figure 2.12 The flow diagram of the proposed anomaly detection approach

2.16 Efficient Black-Box Video Adversarial Attacks with GEO-TRAP

In the work by Li, Aich, and Zu [17], the authors focused on addressing the challenges of black-box adversarial attacks against video classification models. The authors highlight the relative lack of attention given to video-based attacks when compared to their image-based counterparts. This disparity is attributed to the added complexities introduced by the temporal dimension in videos, which makes gradient estimation for attacks more challenging. The paper introduces a novel approach called "Geometric TRAnsformed Perturbations" (GEO-TRAP) to effectively search for gradients that maximize the misclassification probability of target videos. The authors also explain that query-efficient black-box attacks rely on accurate gradient estimation to create adversarial examples that lead to misclassification. In videos, the temporal dimension presents challenges in gradient estimation, making the attack process more resource-intensive. To address this, the authors propose GEO-TRAP, an iterative algorithm that leverages geometric transformations to parameterize the temporal structure of the search space. This parameterization reduces the search space and focuses the attack on a small group of parameters that describe the transformations. Experimental results on the widely used Jester dataset demonstrate the superiority of GEO-TRAP. The proposed method achieves better attack success rates with approximately 73.55% fewer queries compared to the current state-of-the-art method for black-box video adversarial attacks. The experimental results underline the method's effectiveness, showcasing its potential for identifying vulnerabilities in video classification models and advancing the understanding of black-box attacks in the video domain.

Figure 2.13 shows an overview of Geo-TRAP

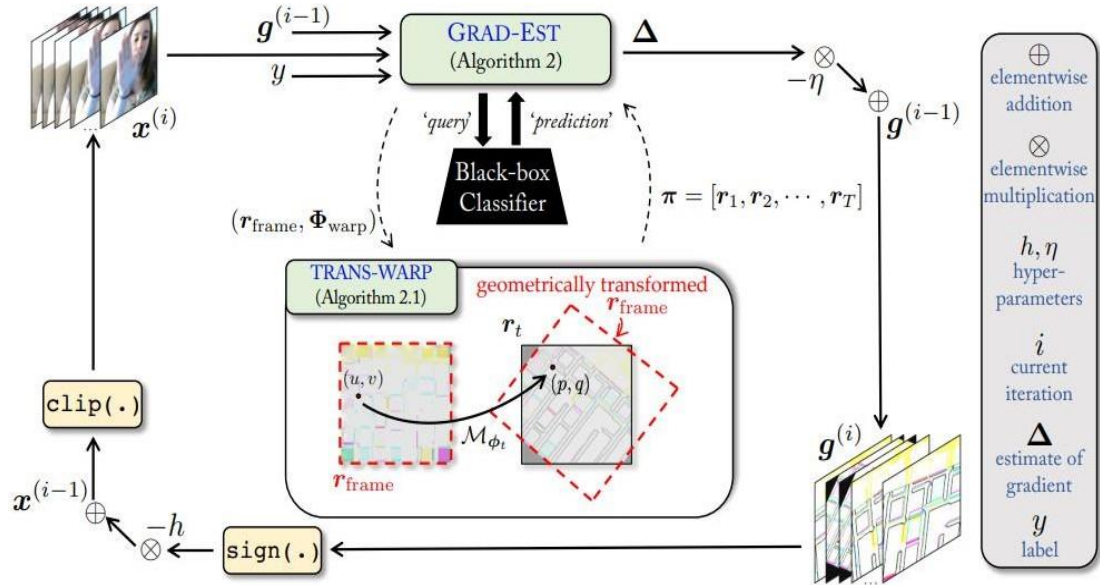


Figure 2.13 Overview of Geo-TRAP

2.17 Enhancing Robustness in Video Adversarial Attacks using V3A Technique

In the work by Yin, Xu, Hu [18], the authors demonstrate the vulnerability of video classification systems to adversarial attacks and propose a novel approach to generating robust video adversarial examples. They recognize that traditional methods of perturbing video inputs with noise may not result in highly robust adversarial examples. The authors identify the challenges associated with crafting robust video adversarial examples, a domain that has received less attention compared to image and audio counterparts. The authors propose a solution called Video-Augmentation-Based Adversarial Attack (v3a) to enhance the robustness of video adversarial examples. The proposed v3a approach addresses these challenges by integrating video augmentation techniques to improve the loss function's efficacy, leading to the generation of more robust adversarial examples. Importantly, v3a also considers the balance between perturbation robustness and human perceptibility. By applying transformations selectively and iteratively, the method enhances robustness without introducing

noticeable perturbations. The authors used the UCF-101 dataset and the long-term recurrent convolutional network (LRCN) model to assess v3a's robustness and effectiveness compared to existing methods. The results demonstrate that v3a significantly improves the fooling rate for both white-box and black-box attack scenarios, outperforming benchmarks such as the sparse adversarial video attack (SA) and the heuristic black-box adversarial video attack (HA). The method's ability to maintain mean absolute perturbation (MAP) within acceptable limits showcases its capability to enhance adversarial example resilience without compromising perceptual quality. This study contributes to understanding and addressing the challenges of robustness in video adversarial examples, highlighting v3a's potential for security-critical applications in video classification systems.

Figure 2.14 shows an overview of video-augmentation-based adversarial attack

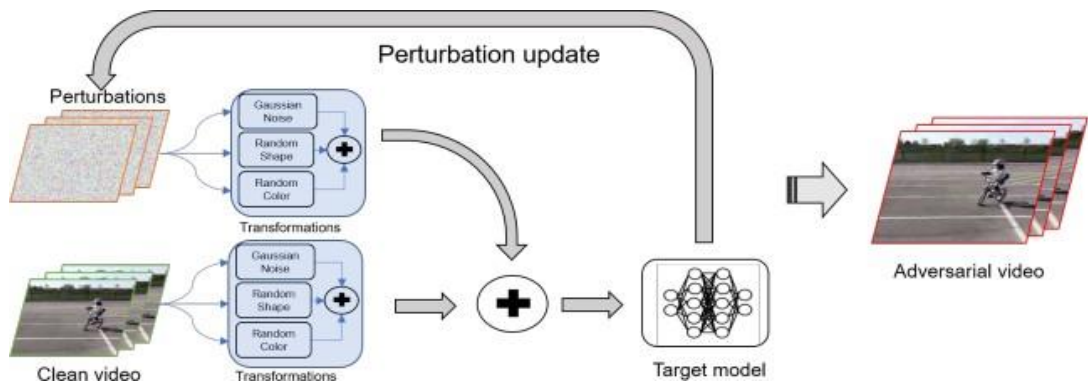


Figure 2.14 Overview of video-augmentation-based adversarial attack

2.18 V-BAD: Black-Box Video Adversarial Attacks and Robustness Evaluation

In the work by Jiang, Ma, Chen [19], the authors introduce a groundbreaking framework that addresses a significant gap in the realm of adversarial attacks by focusing on the generation of black-box adversarial attacks targeting video recognition models by presenting the Video-Based Adversarial Attack (V-BAD) framework. The V-BAD framework harnesses the transferability of adversarial perturbations originating from image models. These perturbations are used as a starting point for the generation of video adversarial examples. To refine and enhance the effectiveness of these perturbations, the framework leverages Natural Evolution Strategies (NES), a

derivative-free optimization technique and its smart application at the patch level of the tentative perturbations. By focusing on patch-level rectification rather than pixel-wise adjustments, the framework achieves remarkable efficiency in generating adversarial gradients. The authors used three prominent video datasets and two state-of-the-art video recognition models to showcase the framework's capability in generating targeted and untargeted adversarial attacks. Notably, V-BAD achieves impressive success rates with a relatively low number of queries to the target models. This efficiency is noteworthy, given that videos inherently possess a significantly higher dimensionality than static images. Consequently, V-BAD emerges as a potent tool not only for generating adversarial attacks but also for evaluating and bolstering the robustness of video recognition models in the face of black-box adversarial attacks. By introducing the concept of black-box video adversarial attacks and presenting the innovative V-BAD framework, the authors have not only tackled an overlooked aspect of adversarial research but also enriched the field with a versatile tool for evaluating and enhancing the security of video recognition models.

Figure 2.15 shows an overview of the proposed V-BAD framework for black-box video attacks

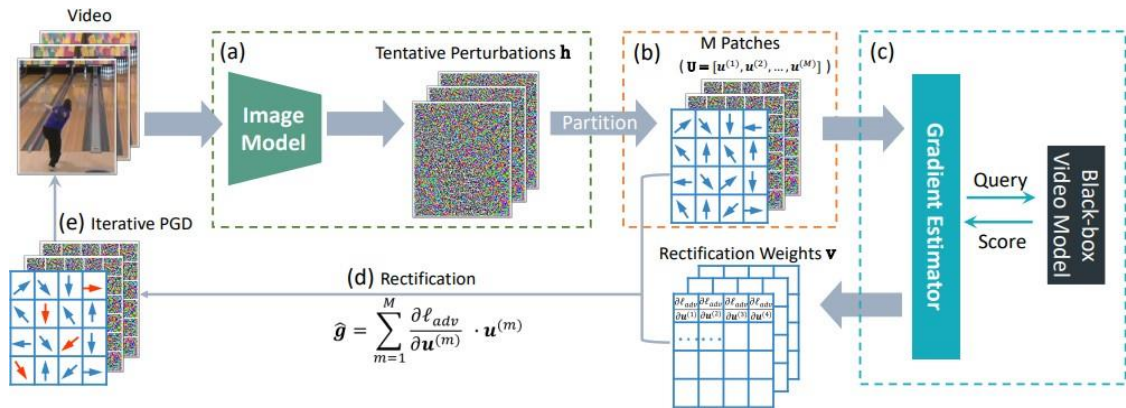


Figure 2.15 Overview of the proposed V-BAD framework for black-box video attacks

In conclusion, this comprehensive literature review illuminates the evolving landscape of adversarial attacks targeting video recognition models. As video-based applications become increasingly integral to critical systems like surveillance, autonomous vehicles,

and video analysis, the susceptibility of these models to adversarial manipulation demands thorough exploration. The unique challenges posed by video sequences, comprising both spatial and temporal intricacies, necessitate tailored approaches for crafting effective adversarial attacks. This review delves into a diverse array of methodologies, strategies, and implications of adversarial attacks in the video domain.

The research examined encompasses a wide spectrum of attack techniques, each targeting specific vulnerabilities of video recognition models. Spatial attacks subtly manipulate individual frames to deceive models' perceptions, while temporal attacks exploit the sequential nature of videos to disrupt motion and scene understanding. The landscape of defenses and countermeasures is also explored, highlighting the ongoing struggle to establish robust defenses capable of withstanding sophisticated adversarial manipulation.

Chapter 3:

Datasets and Models

In recent years, video recognition models have become really good at understanding videos, and they're used in almost all fields of life such as surveillance, recommendation services, self-driving cars, and even healthcare. They help us make decisions based on what we see in real-time. But there's a problem we need to address - something called "adversarial attacks." These attacks are like tricks played on these models. They try to confuse the model by making small changes to the videos it sees. While people have studied these tricks a lot for pictures, it's different when it comes to videos. This chapter is all about how important it is to have the right kind of video data and the right models to deal with these tricky attacks. The main purpose of this research is to find the unusual things happening in a series of video frames. To do this, we need videos where someone has already marked which parts are normal and which parts are abnormal. We used two well-known sets of videos for this task.

3.1 Dataset

We demonstrate the effectiveness of our attack on CrimesScene dataset [\[21\]](#).

3.1.1 Hockey Fight Dataset

The Hockey Fight dataset comprises a collection of 1000 videos, each with an average duration of 1.64 seconds and a frame rate of 25 frames per second.

Among these videos, an equal distribution is maintained, where 50% represent normal instances, while the remaining 50% are designated as fight scenes. The fight scenes are recognized as abnormal instances in this research.

3.1.2 CrimesScene Dataset

We used a new dataset called the "CrimeScene Dataset" [21] for the evaluation of scene recognition task in videos. We got the idea from the UCF Crimes dataset, which has thousands of videos spread over 13 different types of crimes. To detect major crimes in video sequences, we gathered a subset of videos from the UCF Crimes dataset, which includes three categories: fighting, shootings, and vandalism. We carefully marked which parts of these videos were normal and which were abnormal. This labeling helps train models to recognize scenes accurately. We named this new dataset the "CrimesScene Dataset."

Our dataset contains 286 videos obtained from real world CCTV footage in which the average length of videos is 46.9 seconds. There are 201 videos in the training dataset and 85 videos in testing dataset. The normal class is labeled as 0 whereas abnormal is labeled as 1. All these details are shown in the table below.

Dataset	Classes	Videos	Normal Scenes	Abnormal Scenes
Hockey fight	Fighting	Single	500	500
	Fighting	45	1045	1962
CrimesScene [Ours]	Shooting	50	827	889
	Vandalism	51	557	1318
	Total	150	2429	4169

The main contribution of the CrimesScene dataset lies in the improved annotation quality for the three categories within the UCF Crimes dataset. We have provided a finely annotated dataset for real-world major crimes, making it suitable for supervised learning-based algorithms. Figure 3.1 shows the procedure adapted to annotate frames of video sequence in Normal and Abnormal category. This activity is repeated for videos of each class.

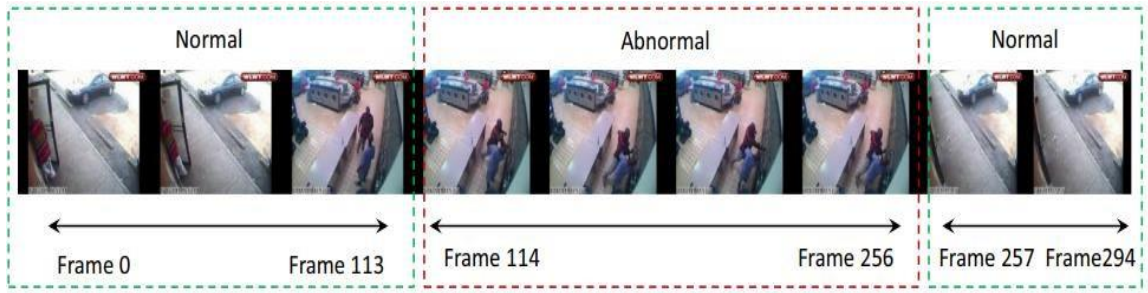


Figure 3.1 The procedure adapted to annotate frames of video sequence in Normal and Abnormal category. This activity is repeated for videos of each class

We've created a highly detailed dataset for significant real-world crimes, making it ideal for training supervised learning algorithms. This dataset includes videos that fall into three similar categories of major crimes. Within each category, there's an imbalance between normal and abnormal scenes. We've collected scenes from about 50 different videos for each crime type, which means there's a variety of environmental conditions and situations. Figure 3.2 (a), (b), (c) and (d) shows the instances from the normal, fighting, shooting, and vandalism classes of the Crimesscene dataset.



Figure 3.2 Normal instances from the CrimesScene dataset



(a) Fighting class from the CrimesScene dataset



(b) Shooting class from the CrimesScene dataset



(c) Vandalism class from the Crimesscene dataset

Due to these characteristics, this dataset demands advanced deep learning techniques to effectively extract features and identify abnormal scenes within video sequences.

3.2 Video Recognition Models

3.2.1 Convolution 3D Block

A 3D convolutional block [33], often referred to as a "3D Convolution Block" or simply "Conv3D Block," is a fundamental component in 3D convolutional neural networks (CNNs). It's used for processing three-dimensional data, typically applied to video data or volumetric data such as medical images or 3D scans. This block helps extract hierarchical features from the input data.

A typical 3D Convolutional Block consists of the following layers:

1. 3D Convolutional Layer: This layer applies a set of learnable filters to the input data, just like a 2D convolutional layer in traditional CNNs but in three dimensions. These filters slide over the input volume, computing convolutions along the spatial dimensions (width, height) and the temporal dimension (time or depth).
2. Activation Function: After each convolution operation, an activation function is applied element-wise to introduce non-linearity into the network. Rectified Linear Unit (ReLU) is a common choice.

3. **Batch Normalization:** Batch normalization is used to stabilize and speed up training by normalizing the activations of the previous layer within a mini-batch.
4. **Pooling Layer:** This layer performs down-sampling, reducing the spatial and temporal dimensions of the feature maps while keeping the most important information. MaxPooling3D, for example, selects the maximum value from a small 3D region.

These components are typically stacked together to create a Conv3D Block, and multiple Conv3D Blocks are often used in sequence to build deep 3D convolutional neural networks for tasks like video classification, action recognition, or medical image analysis.

The exact architecture of a Conv3D Block can vary depending on the specific neural network architecture and the task at hand, but the core idea remains the same: applying 3D convolutions to capture spatiotemporal features in three-dimensional data.

The Conv3B Block has achieved state-of-the-art performance in various fields, but they require a large amount of training data and are computationally expensive.

A simple Conv3D block is shown in the Figure 3.3.

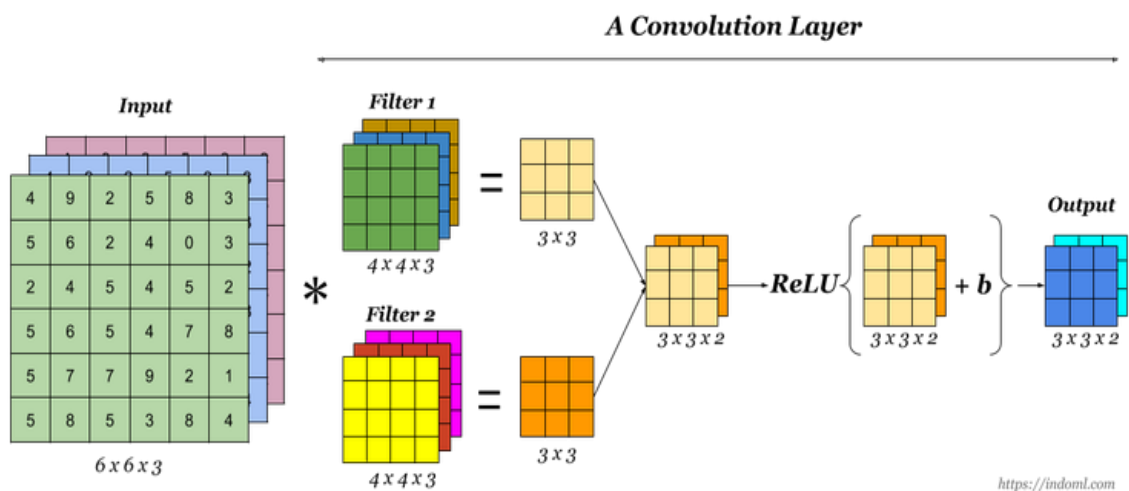


Figure 3.3 A simple Conv 3D Block

3.2.2 Pseudo 3D Block

A pseudo 3D block, sometimes referred to as a "Pseudo 3D Convolution Block," is a specialized building block commonly used in two-stream convolutional neural networks (CNNs) for video analysis tasks [\[21\]](#). It's called "pseudo" because it simulates a three-dimensional convolutional operation while actually performing separate 2D convolutions on two different input streams: one for spatial information and one for temporal information.

Here's how a typical pseudo 3D block works:

1. **Spatial Stream:** The spatial stream takes in individual frames (2D images) from a video sequence. It processes each frame using a standard 2D convolutional layer ($3 \times 3 \times 1$), which captures spatial features within each frame and encodes appearance information.
2. **Temporal Stream:** The temporal stream takes multiple frames (typically a short sequence of consecutive frames) as input. It applies a 1D convolutional layer ($1 \times 1 \times 3$) along the temporal dimension (time) to capture temporal patterns and motion information across frames.
3. **Fusion:** After processing both streams separately, the results are fused or combined in some way. This fusion can take different forms, such as concatenating the feature maps or using element-wise operations like addition or multiplication. The idea is to merge spatial and temporal information.
4. **Activation Function:** An activation function like ReLU is applied to introduce non-linearity.
5. **Batch Normalization:** Batch normalization may be applied to stabilize and speed up training.

By using two separate streams—one for spatial and one for temporal information—and combining their outputs, pseudo 3D blocks aim to capture both spatial and temporal features in video data. This approach is computationally efficient compared to full 3D convolutions while still achieving good performance in video analysis tasks like action recognition and video classification.

Overall, pseudo 3D blocks are an important component of two-stream CNN architectures designed for video-related tasks, allowing models to effectively process spatiotemporal information. They leverage the power of 2D convolutional neural networks and requires less memory than 3D convolutional networks. Their only drawback is that they operate on temporally segmented data and hence may not capture long-term temporal dependencies as well as 3D convolutional networks, which operate on the entire temporal sequence

Three different variants of P3D units P3DA, P3DB, and P3DC were developed for considering direct and indirect influence between the two filters. Figure 3.4 shows different variations of the Pseudo-3D Block.

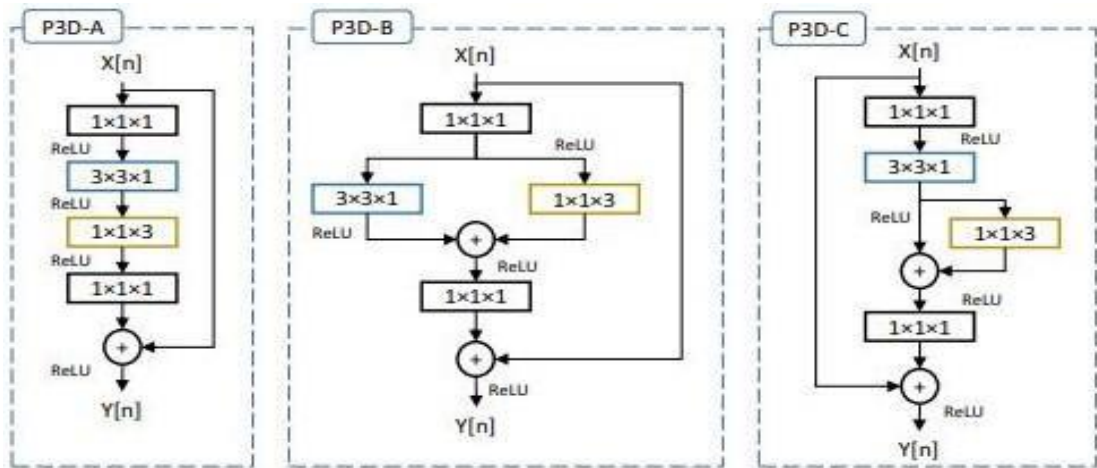


Figure 3.4 Different variations of the Pseudo-3D Blocks

3.2.3 Quasi 3D Block

A quasi-3D block, also known as a "Quasi-3D Convolutional Block," is a building block used in deep learning architectures, primarily for video analysis tasks. It is a variant of the Pseudo-3D block and combines both 2D and 3D convolutional operations to capture spatial and temporal features in video data effectively. [21]

Here's how a quasi-3D block typically works:

1. **Spatial Stream (2D Convolution):** The spatial stream processes individual frames or images from a video sequence using a standard 2D convolutional

layer. This layer captures spatial features within each frame, like how traditional 2D convolutional neural networks (CNNs) operate.

2. Temporal Stream (3D Convolution): The temporal stream takes a short sequence of consecutive frames (video clips) as input and applies a 3D convolutional layer along both the spatial and temporal dimensions. This 3D convolution captures temporal patterns and motion information across frames.

It uses a $3 \times 1 \times 3$ filter to encode the temporal variation along the horizontal axis.

It uses a $1 \times 3 \times 3$ filter to encode the temporal variation along the vertical axis.

3. Fusion: After processing both the spatial and temporal streams, their outputs are fused or combined in some way. This fusion step typically involves concatenating the feature maps from both streams along a certain dimension or using element-wise operations like addition or multiplication. The goal is to merge the spatial and temporal information effectively.
4. Activation Function: An activation function, such as the Rectified Linear Unit (ReLU), is applied to introduce non-linearity.
5. Batch Normalization: Batch normalization may be applied for better training stability and faster convergence.

The quasi-3D block leverages both 2D and 3D convolutional operations, allowing it to capture spatial details within individual frames and temporal dynamics across frames simultaneously. This approach strikes a balance between computational efficiency and performance, making it well-suited for video-related tasks like action recognition, video classification, and spatiotemporal feature extraction.

In summary, the quasi-3D block is a key component in architectures designed for video analysis, as it efficiently extracts both spatial and temporal features from video data, which is crucial for understanding and recognizing actions and events in videos. It reduced the number of parameters and computational complexity of the model and also achieved comparable performance to traditional 3D convolutional networks on video analysis tasks.

There are three different variants of the Q3D block in terms of their architectural differences.

1. Q3D-A: In this design, we have a series of filters that are connected in a cascade manner. Specifically, we first apply a spatial filter (S), followed by a temporal filter (T). Then, we have a spatiotemporal filter in the X direction ($T\text{\$}$) and another one in the Y direction ($T\text{\dagger}$), in that particular sequence. In this architecture, these filters have a direct influence on each other as they follow a common path or sequence. In other words, the output of one filter directly affects the input or behavior of the next filter in the chain.
2. Q3D-B: This design is created by arranging the filters in a parallel configuration to enable an indirect influence among them. These filters are connected along separate pathways, all sharing the same input and collectively contributing to the output. In this representation, these elements and filters are interconnected in a way that allows them to jointly affect the final output, with each filter operating in parallel and contributing to the overall result.
3. Q3D-C: This design is implemented to establish a direct connection among the spatial and temporal filters, along with the block's output, aiming to capture the combined effect of all spatial and temporal dimensions. To achieve this, two additional 1D filters, denoted as $S\text{\$}$ and $S\text{\dagger}$, have been introduced in this modified version of the Quasi-3D (Q3D) block. In this representation, these elements and filters work together to directly influence one another, allowing for the comprehensive integration of spatial and temporal dimensions in the output.

The different variations of the Q3D block are shown in Figure 3.5 below.

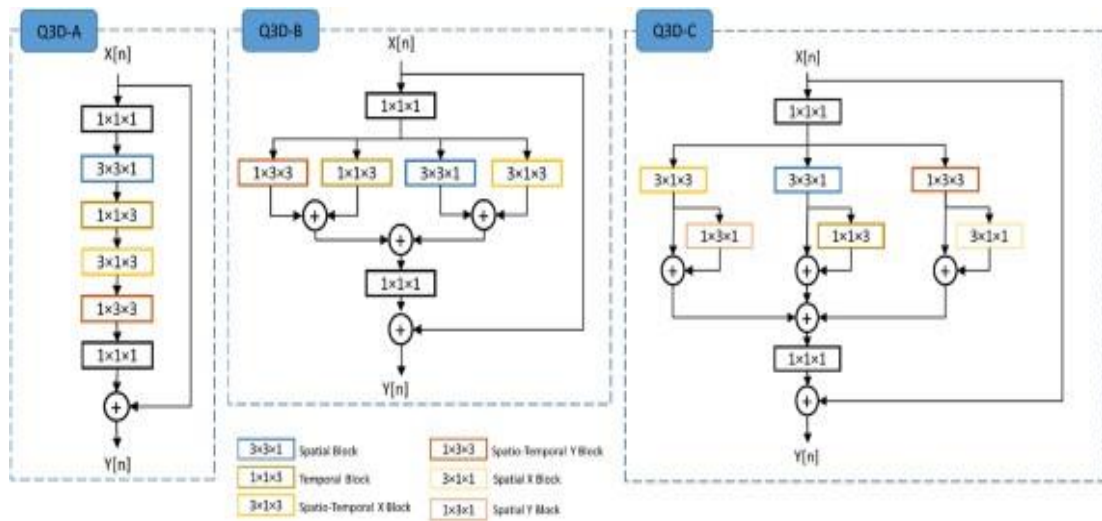


Figure 3.5. Different variations of the Q3D block

In this chapter, we delve into the pivotal role of datasets in advancing our understanding of adversarial attacks in the context of video recognition models. As we explore the characteristics, creation, and utilization of video datasets, we gain insights into the unique challenges posed by the temporal dimension of videos. By curating and utilizing robust video datasets, we enhance the security and resilience of video recognition models in an ever-evolving landscape of adversarial threats. This chapter serves as a critical foundation for the subsequent exploration of adversarial techniques and their impact on video-based AI systems.

Chapter 4

Designing Adversarial Attacks

This chapter introduces different methods of designing adversarial attacks on videos. Our goal is to explore various methods for creating attacks specifically targeted at video recognition models. These models are essential in applications like surveillance, autonomous vehicles, and content recommendation systems. However, their vulnerability to subtle manipulations in input data is a significant concern that needs thorough investigation.

In this exploration, we will cover a range of adversarial attack techniques, each with its unique approaches and strategies. This includes creating perturbations that can fool video recognition models and the art of selecting keyframes and timing disruptions strategically. Section 4.1 to 4. lists the different methodologies designed for performing adversarial attacks on our video recognition models.

4.1 Adversarial Attacks on Video Recognition Models using Adversarial Patch Technique

Our first approach takes into account a rather intriguing and effective form of adversarial attack using an "Adversarial Patch." [\[22\]](#) This attack method involves adding a carefully crafted, seemingly harmless patch or image overlay to an existing scene or object. The goal is to manipulate the model's perception of the scene or object to produce a desired misclassification or erroneous interpretation.

4.1.1 Methodology

The adversarial patch method involves a systematic approach to create and apply patches that can deceive video recognition models [22]. These patches are illustrated in Figure 4.1 and they are chosen on the basis of certain key parameters. Those key parameters include.

- **Robustness:** These adversarial patches are designed to withstand variations in lighting, angle, scale, and other environmental factors. This robustness ensures that the attack remains effective across diverse conditions, making it a valuable tool for real-world scenarios.
- **Universality:** These patches are universal as they are trained on a large and diverse dataset, ensuring that the patch remains effective across various scenarios and models.
- **Targeted Attacks:** These adversarial patches can be crafted to target specific objects or classes within an image. This level of specificity allows attackers to manipulate model behavior with precision, potentially leading to misclassifications that have real-world implications.

First, we select the frames of our video sequence that we want to manipulate. After that, we load any one image (out of the three shown in Figure 4.1) that is likely intended to be used as an adversarial patch.



Figure 4.1 (a) Toaster-Target (b) Crab-Target (c) Toaster-Target (Disguised).

After loading the patch, we begin by specifying the location in our target frame where the patch should be placed. The available options include the top right corner, bottom right corner, top left corner, and bottom left corner. After that we resize it to a smaller

dimension so that the patch matches the dimensions of frames or objects in the target video where the patch will be applied. This step ensures that the patch fits appropriately within the video frames. After that, we convert the patch image from the BGR color space to the RGB color space to ensure that the color representation of the patch matches the expected color format in the video frames. This is done to avoid the differences between the image and the patch i.e., making it imperceptible to the human eye. After this, the resized and color-converted adversarial patch will be overlaid onto selected frames of the target video and fed to our video recognition models leading to misclassifications or misinterpretations by the models. This can be formulated using Algorithm 4.1.

Algorithm 4.1 Attack through Adversarial Patch

```

1: Load initial video sequence  $V$ .
2: Load target video sequence  $\hat{V}$ .
3: Load adversarial patch image as patch.
4: Resize patch image.
function applyAdversarialPatch( $V$ ,  $\hat{V}$ , patch, patch_location):
5: Initialize empty video sequence  $V^*$ .
6: for each frame  $F$  in  $V$  do
7: FrameWithPatch  $\leftarrow$  applyPatch( $F$ , patch, patch_location)
8: append FrameWithPatch to  $V^*$ .
9: end for
10: return  $V^*$ .
function applyPatch(frame, patch, patch_location):
11: Initialize empty frame outputFrame.
12: Copy content of frame to outputFrame.
13: Paste adversarial patch onto outputFrame at patch_location.
14: Return outputFrame.
15: Load  $V$ ,  $\hat{V}$ , the adversarial patch, and its location (patch_location).
16:  $V^* \leftarrow$  applyAdversarialPatch( $V$ ,  $\hat{V}$ , patch, patch_location).
17: Save  $V^*$ .

```

Figure 4.2 (a) and (b) shows the original and perturbed frame after adding the adversarial patch in the top left corner.

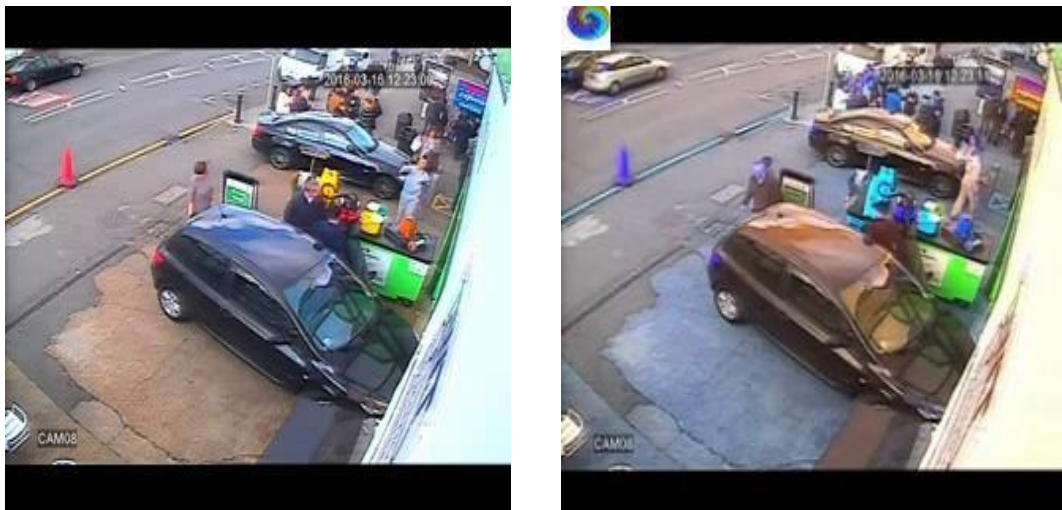


Figure 4.2 (a) Original frame (b) Perturbed frame after adversarial patch

4.1.2 Consequences and Implications

Adversarial patches pose significant security and privacy risks across various domains. They can deceive surveillance and facial recognition systems, enabling unauthorized access and evasion of surveillance. In the realm of autonomous vehicles, these patches can jeopardize road safety by causing misinterpretations of the environment. Content moderation on platforms like social media becomes vulnerable, as malicious content can evade detection. Furthermore, recommendation systems may provide misleading suggestions due to attacks on user preference perception, ultimately impacting user experiences and content consumption.

4.2 Appending Adversarial Frames to Video Sequences for Adversarial Attacks

The second approach is the appending of adversarial frames after several frames of a video [13]. This technique tampers the temporal consistency of the videos thus holding the potential to disrupt video-based machine learning models and poses significant challenges to the integrity of digital video content.

4.2.1 Methodology

The process of appending adversarial frames to a video sequence involves strategically adding frames to the existing content, which subtly incorporate adversarial perturbations and deceive machine learning models. These frames are generated using adversarial attack algorithms, which often require knowledge of the target model's architecture and parameters. The main objective is to introduce adversarial content to the video sequence while allowing for the adjustment of FPS to observe varying results.

In the first step, we begin by selecting an initial FPS setting for the video sequence. This value will serve as the baseline for the video manipulation process. For the sake of simplicity, we have chosen to maintain the initial FPS at a standard rate of 30 frames per second (FPS) commonly used in video content. In this context, the value of FPS will signify the frequency at which an adversarial frame is inserted into the video sequence, indicating the number of original frames that precede each insertion.

Next, we generate adversarial frames for the attack using techniques like generative adversarial networks (GANs) or gradient-based methods. These frames should be designed to deceive AI models while remaining visually inconspicuous to human observers. In this case, we have employed a carefully crafted adversarial image, as illustrated in Figure 4.1. In the next step, we strategically place the adversarial frames at specific points within the video sequence to maximize their impact i.e we have inserted it into video sequences at a rate of one frame per second (1 FPS) while maintaining the temporal coherence in the video by considering the flow of events. Inserted frames should not disrupt the natural progression of the video. This technique is designed to exploit the temporal nature of videos, where the adversarial frames emerge after several frames, making them nearly imperceptible to human observers. The deliberate timing of these insertions aims to deceive human perception, as the alterations become progressively subtle and inconspicuous. [\[13\]](#)

After that, we fed the altered video sequence to our video recognition models and assessed the success of the attack by measuring its impact on the performance of those models. Then we iterate through the process, adjusting the parameters, including FPS, adversarial perturbation magnitude, and interpolation techniques, to optimize the

results. The goal is to find the right balance between adversarial effectiveness and human perceptibility.

This approach poses a significant challenge to the human eye's ability to detect subtle visual discrepancies, potentially leading to a false sense of authenticity in the manipulated video content. This can be formulated using Algorithm 4.2.

Algorithm 4.2 Attack through Appending Adversarial Frames

Require: Functions for Adversarial Frame Insertion **function** append Adversarial Frames(**video**, **adversarial_frames**, **insertion_rate**, **start_fps**, **end_fps**):

```

1: Initialize an empty video sequence for the manipulated video.
2: manipulated_video  $\leftarrow$  []
3: current_fps  $\leftarrow$  start_fps
4: frame_counter  $\leftarrow$  0
5: for each frame F in video do
6:   append F to manipulated_video
7:   if frame_counter % current_fps == 0:
8:     adversarial_frame  $\leftarrow$  adversarial_frames.pop()
9:     append adversarial_frame to manipulated_video
10:  if current_fps > end_fps:
11:    current_fps  $\leftarrow$  current_fps - 1
12:    frame_counter  $\leftarrow$  frame_counter + 1
13:  return manipulated_video
14: original_frames  $\leftarrow$  loadFrames("original_frames_directory/")
15: adversarial_frames  $\leftarrow$  loadAdversarialFrames("adversarial_frames_directory/")
16: insertion_rate  $\leftarrow$  1 # Insert adversarial frames every 1 second (1 FPS).
17: start_fps  $\leftarrow$  30
18: end_fps  $\leftarrow$  10
19: manipulated_frames  $\leftarrow$  appendAdversarialFrames(original_frames,
adversarial_frames, insertion_rate, start_fps, end_fps)
20: saveFramesAsVideo(manipulated_frames,
"manipulated_video_with_adversarial_frames.mp4")

```

Figure 4.3 (a), (b), and (c) shows the original frame, frame after 3 fps and frame after 5 fps in which the adversarial patch is appended after 5 FPS.



Figure 4.3 (a) Original Frame (b) After 3 FPS (c) After 5 FPS

4.2.2 Challenges

The appending of adversarial frames within video sequences poses a potent and evolving threat to the fields of computer vision and cybersecurity. These imperceptible alterations can compromise the security of systems relying on computer vision, including surveillance cameras, facial recognition, and autonomous vehicles, by deceiving them into making incorrect decisions, potentially leading to security breaches or accidents. Moreover, the technique has profound implications for the dissemination of disinformation, as malicious actors can manipulate news broadcasts, social media videos, or security footage, eroding trust and propagating false narratives in society. As machine learning applications in video-based contexts continue to proliferate, the risk of adversarial attacks disrupting and compromising these systems becomes increasingly significant. Thus, understanding the methodology, challenges, and consequences of appending adversarial frames is imperative for the development of effective defense

mechanisms and the preservation of the integrity of digital video content in our interconnected world. This research sheds light on the intricacies of adversarial attacks in video data and highlights the importance of developing robust countermeasures in the field of computer vision and cybersecurity.

4.3 Adversarial Attacks on Video Recognition Models through Gaussian Noise

4.3.1 Introduction to Gaussian Noise

Gaussian noise, sometimes called white noise, is a key idea in signal processing and statistics. It's a kind of random pattern that follows a familiar bell-shaped curve, known as the Gaussian distribution, as shown in Figure 4.2. This noise looks like a bell curve and has a few key features: it has a zero mean and unit standard deviation, each piece of noise is independent, and it's got a uniform amount of energy across different frequencies. Because of these characteristics, Gaussian noise is used in different areas like electronics, communication, and image editing. It helps describe the natural randomness or errors that can show up in data and signals. [25] Knowing and dealing with Gaussian noise is essential for tasks like cleaning up data or making accurate measurements.

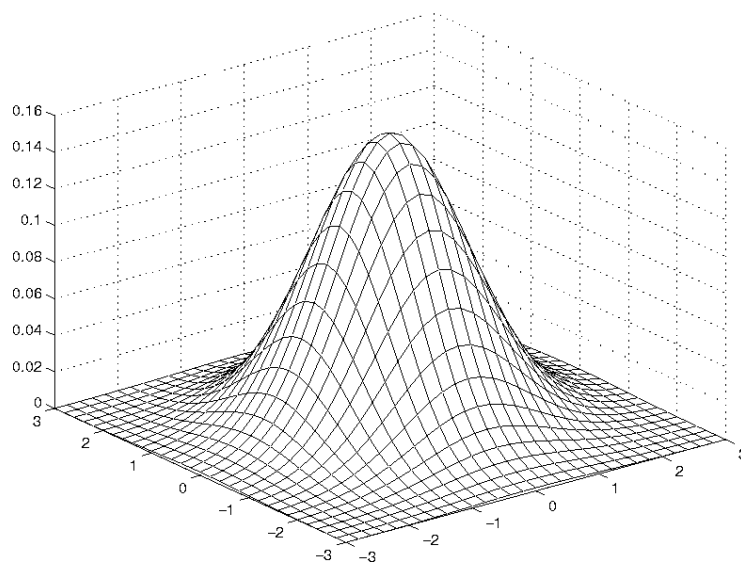


Figure 4.2 Gaussian Distribution

4.3.2 Adversarial Attacks using Gaussian Noise

Adversarial attacks on video recognition models using Gaussian noise involve adding subtle noise to individual frames of a video sequence. The noise is drawn from a Gaussian distribution, making it appear as random variations that are often imperceptible to the human eye but capable of misleading the model's predictions.

In our experiments, we have added subtle Gaussian noise to each frame of our video sequence which disrupts the visual coherence of video frames, leading our video recognition models to misclassify or misinterpret the content. The carefully crafted noise perturbations cause the model to make incorrect decisions, which is particularly concerning in safety-critical applications like autonomous vehicles.

4.3.2.1 Methodology

In our approach to these attacks, we uniformly introduced Gaussian noise across all frames of the video sequence. Initially, we determined the specific characteristics of the Gaussian noise by setting its mean and standard deviation. Combining mean and standard deviation allows to craft noise patterns that not only introduce bias (mean) but also control the level of randomness or unpredictability (standard deviation) in the perturbations.[\[26\]](#) Low values suggest subtler attacks, while high values indicate more pronounced and potentially malicious attacks aimed at either causing misclassifications or obscuring the visual content of the video. While the mean typically remained at zero, we also conducted evaluations with adjusted mean values as needed. For each individual frame within the video, we generated Gaussian noise samples based on the predefined mean and standard deviation. To achieve this, we employed a random number generator that adheres to the Gaussian distribution, thus creating a set of random noise values.

Subsequently, we seamlessly incorporated this generated Gaussian noise into each frame of the video sequence. This incorporation was achieved by straightforwardly adding the generated noise values to the pixel values of each frame. To ensure that the pixel values remained within the valid range (e.g., 0 to 255 for 8-bit images), we applied clipping to the noisy pixel values.

Fine-tuning the noise levels was possible by adjusting the standard deviation. Higher standard deviations were associated with more pronounced noise, resulting in noisier frames, while lower values yielded a milder noise effect. Finally, after successfully introducing Gaussian noise to all selected frames, we preserved the altered video frames for subsequent analysis and experimentation. This can be formulated using Algorithm 4.3.

Algorithm 4.3 Attack through Gaussian Noise

Require: Input image \mathbf{I} , Mean ($\boldsymbol{\mu}$), Standard Deviation ($\boldsymbol{\sigma}$)

- 1: Create empty `noisy_image`
 - 2: **for** each pixel (\mathbf{x}, \mathbf{y}) in \mathbf{I} **do**
 - 3: Generate random noise $\delta \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$
 - 4: Apply: $I(x, y) = I(\mathbf{x}, \mathbf{y}) + \delta$
 - 5: Clip: $I(x, y) = \text{clip}(I(\mathbf{x}, \mathbf{y}), \mathbf{0}, \mathbf{255})$
 - 6: `noisy_image(x, y) = I(x, y)`
 - 7: **end for**
 - 8: **return** `noisy_image`
-

Figure 4.4 (a) and (b) shows the original and perturbed image after the addition of Gaussian noise.



Figure 4.4 (a) Original Image (b) Perturbed Image after Gaussian Noise

4.3.3 Consequences and Implications

Attacks that use Gaussian noise on video recognition models pose two main problems: security and privacy. On the security side, these attacks can make the models make mistakes, which could let unauthorized people in or miss important security issues in things like surveillance. On the privacy side, these attacks can lead to misunderstandings by the models, potentially causing unnecessary spying or exposing people's private information. In simple terms, these attacks are a growing threat to the reliability and safety of computer vision systems, putting security and privacy at risk by confusing the models that rely on how things look and move in videos.

4.4 Adversarial Attacks on Video Recognition Models Through Contrast Adjustment

The fourth approach of performing adversarial attacks involves the manipulation of video data through contrast adjustment. This technique aims to exploit the shortcomings of video recognition systems by subtly altering the contrast levels within video frames [27]. Through this attack, we dive deeper into the intricacies of adversarial attacks using contrast adjustment on video recognition models.

4.4.1 Methodology:

In this methodology, we adjusted the contrast across all frames of our video sequence for manipulating the video sequence. First, we adjusted the parameters for contrast adjustment by specifying the alpha (α) and beta parameters. The parameter alpha (α) corresponds to a scaling factor that controls the contrast adjustment. It determines how much to increase or decrease the contrast in an image. When alpha (α) is set to 1.0 (or 100%), it means no change in contrast. The image remains as it is. When alpha (α) is greater than 1.0, it increases the contrast. This makes the dark areas of the image darker and the bright areas brighter, leading to a higher overall contrast. When alpha (α) is less than 1.0, it decreases the contrast. This has the opposite effect, making dark areas lighter and bright areas darker, resulting in reduced contrast. By adjusting `alpha`, we effectively control the extent of contrast modification. The parameter beta (β) allows us to shift the brightness of the image. When beta (β) is zero, there is no brightness shift. If beta (β) is set to a positive value, it increases the brightness of the entire image. If beta (β) is set to a negative value, it decreases the brightness of the entire image.

The alpha (α) and beta (β) parameters provide fine-grained control over how the contrast and brightness of an image are adjusted. We aim to adjust these parameters such that the alterations blend seamlessly into the video stream, making them difficult for human observers to notice. [28] To ensure that the pixel values remained within the valid range (e.g., 0 to 255 for 8-bit images), we applied clipping to the altered pixel values.

Finally, the altered frames (contrasted frames) can be fed into recognition models, potentially causing them to make incorrect decisions because these contrasted frames significantly impact how recognition models perceive and interpret visual content within videos, ultimately resulting in misclassification or misinterpretation. This can be formulated using Algorithm 4.4.

Algorithm 4.4 Attack through Contrast Adjustment

Require: Initial video sequence \mathbf{V} and target video sequence $\hat{\mathbf{V}}$

function adjustContrast(\mathbf{V} , $\hat{\mathbf{V}}$, α , β):

- 1: $\mathbf{V}^* \leftarrow []$ // Initialize an empty video sequence.
- 2: **for** each frame F in \mathbf{V} **do**
- 3: AdjustedFrame \leftarrow applyContrastAdjustment(F , α , β)
- 4: append AdjustedFrame to \mathbf{V}^*
- 5: **end for**
- 6: **return** \mathbf{V}^*

function applyContrastAdjustment(**frame**, α , β):

- 7: outputFrame $\leftarrow []$ // Initialize an empty frame.
- 8: **for** each pixel (x, y) in frame **do**
- 9: adjustedPixel \leftarrow (frame(x, y) * α) + β
- 10: adjustedPixel \leftarrow clip(**adjustedPixel**, 0, 255)
- 11: **append** adjustedPixel to outputFrame
- 12: **end for**
- 13: **return** outputFrame

function clip(**value**, min_value, max_value):

- 14: **if** value < min_value, value \leftarrow min_value
 - 15: **if** value > max_value, value \leftarrow max_value
 - 16: Return value
 - 17: Load \mathbf{V} , $\hat{\mathbf{V}}$, α , and β
 - 18: $\mathbf{V}^* \leftarrow$ adjustContrast(\mathbf{V} , $\hat{\mathbf{V}}$, α , β)
 - 19: Save \mathbf{V}^*
-

Figure 4.5 (a) and (b) shows the original and perturbed image after the contrast adjustment.



Figure 4.5 (a) Original Image (b) Perturbed image after contrast adjustment

4.4.2 Consequences and Implications

Adversarial attacks using contrast adjustment carry significant implications across security, safety, and privacy domains. They pose a severe security risk by potentially deceiving surveillance systems, leading to missed threats or false alarms in situations where precise video analysis is imperative. They can also lead to certain safety concerns particularly in autonomous vehicles, where altered road signs or obstacles due to contrast adjustment can confuse perception systems, increasing the risk of accidents or unsafe driving conditions. Additionally, privacy becomes compromised as attackers manipulate video content to reveal sensitive information or distort identities in surveillance footage. These attacks primarily aim to influence recognition models, as detailed in the methodology, potentially causing misclassifications or misinterpretations by feeding contrast-adjusted frames into models, further highlighting the profound consequences of such adversarial tactics. As recognition models' importance grows, comprehending and mitigating these attacks is crucial to ensure the reliability and security of computer vision systems in our interconnected world.

4.5 Adversarial Attacks on Video Recognition Models using Salt and Pepper Noise

The fifth technique involves performing adversarial attacks using salt and pepper noise, a powerful technique to compromise the accuracy and reliability of video recognition systems. [29] Though this noise appears to be seemingly harmless, it significantly impacts the reliability and security of video recognition systems.

4.5.1 Introduction to Salt and Pepper Noise

Salt and pepper noise, also known as impulse noise, is random interference affecting digital images and videos. It appears as scattered white and black pixels resembling salt and pepper grains, often due to transmission errors or sensor glitches. This noise degrades image quality, making affected regions appear speckled or grainy. To mitigate its impact, image processing techniques like median filtering or averaging are employed, replacing noisy pixels with values consistent with their surroundings to restore image quality and reduce visual artifacts.[30] Figure 4.2 (a) and (b) shows the impact on an image before and after adding salt and pepper noise, respectively.



Figure 4.2 (a) Before adding salt and pepper noise (b) After adding salt and pepper noise

4.5.2 Methodology

In this method, we randomly inject white and black pixels (salt and pepper) into video frames at strategic locations. We first select our video frames that we want to alter and

then define the parameters for adding salt and pepper noise. The parameters are `salt_prob`` and `pepper_prob``, which represent the probability of adding salt (white pixels) and pepper (black pixels) noise, respectively. Before performing any attack, we create a copy of the video frame to avoid altering the original and calculate the total number of pixels in the frame. Based on the defined probabilities (`salt_prob`` and `pepper_prob``), we compute the number of salt and pepper pixels to be added which is done by multiplying the total pixel count by the respective probabilities.

Then, we generate random coordinates within the frame for adding salt noise. The number of salt pixels is determined by `num_salt``. Set the pixel values at these coordinates to the maximum intensity (255 in the code), creating white pixels (salt) in the frame.

After that, we generate random coordinates for adding pepper noise. The number of pepper pixels is determined by `num_pepper``.[\[30\]](#) Set the pixel values at these coordinates to the minimum intensity (0 in the code), creating black pixels (pepper) in the frame.

We ensure that the noise addition maintains temporal consistency. We inject the noise across multiple frames, making the noise pattern consistent across the video sequence.

Finally, after adding the Salt and Pepper noise to all frames, we assess the success of the adversarial attack by evaluating how the manipulated video sequence affects the behavior of the recognition model. The model's misclassification rate or its ability to make incorrect decisions is measured based on the manipulated frames. This can be formulated using Algorithm 4.5.

Algorithm 4.5 Attack through Salt and Pepper Noise

Require: Initial video sequence \mathbf{V} and target video sequence $\hat{\mathbf{V}}$

function addSaltAndPepperNoise(\mathbf{V} , $\hat{\mathbf{V}}$, salt_prob, pepper_prob):

```

1:  $\mathbf{V}^* \leftarrow []$  // Initialize an empty video sequence.
2: for each frame  $F$  in  $\mathbf{V}$  do
3: NoisyFrame  $\leftarrow$  applySaltAndPepperNoise( $F$ , salt_prob, pepper_prob)
4: append NoisyFrame to  $\mathbf{V}^*$ 
5: end for
6: return  $\mathbf{V}^*$ 

```

function applySaltAndPepperNoise(frame, salt_prob, pepper_prob):

```

7: outputFrame  $\leftarrow []$  // Initialize an empty frame.
8: totalPixels  $\leftarrow$  total number of pixels in frame
9: numSalt  $\leftarrow$  salt_prob * totalPixels
10: numPepper  $\leftarrow$  pepper_prob * totalPixels
11: for each pixel  $(x, y)$  in frame do
12: if random number  $<$  salt_prob:
13: Set pixel to 255 // Add salt noise
14: else if random number  $<$  salt_prob + pepper_prob:
15: Set pixel to 0 // Add pepper noise
16: else:
17: Keep the original pixel value
18: append pixel to outputFrame
19: end for
20: return outputFrame
21: Load  $\mathbf{V}$ ,  $\hat{\mathbf{V}}$ , salt_prob, and pepper_prob
22:  $\mathbf{V}^* \leftarrow$  addSaltAndPepperNoise( $\mathbf{V}$ ,  $\hat{\mathbf{V}}$ , salt_prob, pepper_prob)
23: Save  $\mathbf{V}^*$ 

```

Figure 4.6 (a) and (b) shows the original and perturbed image after adding salt and pepper noise.



Figure 4.6 (a) Original Image (b) Perturbed image after adding salt & pepper noise

4.5.3 Consequences and Implications

Adversarial attacks using salt and pepper noise on video recognition models have far-reaching consequences. They introduce security vulnerabilities, potentially causing these models to make erroneous decisions in critical scenarios like surveillance and access control systems, thereby risking security breaches. Moreover, these attacks pose substantial safety risks, particularly in autonomous vehicles, where misinterpretations of traffic signs or obstacles could lead to accidents and unsafe driving conditions. Beyond safety and security, adversaries can exploit this technique to manipulate video content for malicious purposes, including spreading misinformation and disinformation, ultimately eroding societal trust. To counter these threats, continuous advancements in defense mechanisms and research are imperative to safeguard computer vision systems and ensure their responsible and secure integration across various domains.

4.6 Adversarial Attacks on Video Recognition Models using Motion Blur Technique

The sixth approach deals with the deployment of motion blur in videos as a potential adversarial attack. [31] The adversarial attacks through motion blur follow a meticulously crafted methodology designed to manipulate the visual content of videos subtly.

4.6.1 Methodology

In this method, we specify the direction, length, and intensity of the blur effect, which will be applied to the frames for performing the desired attack. First, we determine the parameters for motion blur, which dictate the characteristics of the blur effect. The key parameters include.

- The dimensions of the matrix used for applying the blur effect (called the kernel size). A kernel size of 15 means that the motion blur effect will be applied using a square kernel with dimensions 15x15 pixels. A larger kernel size will result in a more pronounced blur effect, as it covers a larger area of the image. Conversely, a smaller kernel size will produce a less intense blur effect. kernel size and orientation.
- The direction in which the motion blur is applied (called the orientation). By default, the motion blur produces horizontal effect meaning that the objects in the video frames will appear blurred in a horizontal direction, as if they were moving from side to side. By adjusting the orientation parameter, we can control the direction of the blur effect. For example, setting the orientation to a specific angle will produce a diagonal motion blur effect.

After this, we apply the motion blur to specific regions within frames by convolving the frame with a motion blur kernel. The kernel simulates the effect of motion on objects within the video, resulting in blurred frames. [\[31\]](#)

After this we feed the manipulated video data, containing motion-blurred frames to our video recognition models lead to misclassifications or misinterpretations by the models. The attack using motion blur is applied in a way that it becomes nearly imperceptible to the human eye, ensuring that the alterations blend seamlessly into the video stream. This can be formulated using Algorithm 4.6.

Algorithm 4.6 Attack through Motion Blur

Require: Initial video sequence \mathbf{V} and target video sequence $\hat{\mathbf{V}}$

function applyMotionBlur(\mathbf{V} , $\hat{\mathbf{V}}$, kernel_size):

1: $\mathbf{V}^* \leftarrow []$ // Initialize an empty video sequence.

2: **for** each frame F in \mathbf{V} **do**

3: BlurredFrame \leftarrow applyBlur(F , kernel_size)

4: **append** BlurredFrame to \mathbf{V}^*

5: **end for**

6: **return** \mathbf{V}^*

function applyBlur(frame, kernel_size):

7: outputFrame $\leftarrow []$ // Initialize an empty frame.

8: motionKernel \leftarrow createMotionKernel(kernel_size)

9: BlurredFrame \leftarrow applyKernel(frame, motionKernel)

10: **return** BlurredFrame

function createMotionKernel(kernel_size):

11: Create a kernel of size (kernel_size, kernel_size) with central row as ones.

12: **Normalize** the kernel elements.

13: **Return** the kernel.

function applyKernel(frame, kernel):

14: **Convolve** frame with the kernel to obtain BlurredFrame.

15: **Return** BlurredFrame

16: Load \mathbf{V} , $\hat{\mathbf{V}}$, and kernel_size

17: $\mathbf{V}^* \leftarrow$ applyMotionBlur(\mathbf{V} , $\hat{\mathbf{V}}$, kernel_size)

18: Save \mathbf{V}^*

Figure 4.7 (a) and (b) shows the original and perturbed image after motion blur.



Figure 4.7 (a) Original Image (b) Perturbed image after motion blur

4.6.2 Consequences and Implications

Adversarial attacks using motion blur on video recognition models have several important consequences. They pose security risks by causing recognition models to make incorrect decisions, especially in situations like surveillance, which could lead to security breaches. There are safety concerns, particularly in autonomous vehicles, where motion-blurred road signs or obstacles might confuse the vehicle's systems, potentially causing accidents. Privacy issues also arise when attackers manipulate videos to reveal sensitive information or hide people's identities in surveillance footage. Finally, these attacks can be used for misinformation campaigns, spreading false information, and causing confusion in society. Overall, they challenge the reliability and security of computer vision systems, underlining the importance of robust defenses and ongoing research to protect against these threats and maintain trust in these technologies.

4.7 Adversarial Attacks on Video Recognition Models using Frame Dropping Technique

The seventh technique deals with performing adversarial attacks on video recognition models using the frame dropping technique. This technique is a form of temporal

perturbation that disrupts the flow of visual information and can be used to create adversarial examples.

4.7.1 Methodology

In this method, we have manipulated the video data by selectively dropping frames to deceive or compromise the performance of video recognition algorithms. We selectively drop frames at specific timestamps or locations. The goal is to disrupt the temporal consistency and make the video recognition model misclassify the action or object in the video. We dropped every 2nd, 4th and 5th frame for an imperceptible attack. The attack using frame dropping is applied in a way that it becomes nearly imperceptible to the human eye, ensuring that the alterations blend seamlessly into the video stream. This can be formulated using Algorithm 4.7.

Algorithm 4.7 Attack through Frame Dropping

Require: Initial video sequence V and target video sequence \hat{V}

Input: Frame drop rate, Frame drop pattern

Output: Adversarial video sequence V^*

function applyFrameDropping(V , **frame_drop_rate**, **frame_drop_pattern**):

```

1:  $V^* \leftarrow []$  // Initialize an empty video sequence for the adversarial attack.
2: frame_count  $\leftarrow 0$  // Initialize the frame count.
3: for each frame  $F$  in  $V$  do
4:   if frame_count is not in the frame_drop_pattern then
5:     append  $F$  to  $V^*$  // Keep the frame if it's not part of the frame drop pattern.
6:   end if
7:   frame_count  $\leftarrow$  frame_count + 1
8:   if frame_count  $\geq$  frame_drop_rate then
9:     frame_count  $\leftarrow 0$  // Reset the frame count after dropping frames.
10:  end if
11: end for
12: return  $V^*$ 

```

4.7.2 Consequences and Implications

The impact of frame dropping attacks on video recognition models is profound, leading to reduced accuracy and jeopardizing the reliability of these models. Through strategic frame removal, these attacks can disrupt a model's ability to correctly identify actions and objects in videos, potentially introducing severe misclassifications. This has significant implications for surveillance and security applications, where misclassified events may trigger inappropriate responses. Additionally, frame dropping poses a concerning security and privacy risk as attackers could exploit this technique to evade video surveillance systems, compromising individual privacy and overall security.

Chapter 5

Experiments & Results

This chapter delves into a series of rigorous experiments conducted using the CrimeScene dataset. Our primary goal is to gain a deep understanding of how video recognition models fare when faced with adversarial attacks, exploring both their strengths and vulnerabilities.

Our approach to experimentation is meticulous and thorough. We commence by establishing a baseline, measuring the accuracy of our video recognition models when tested on the dataset in its original, unaltered form. This baseline serves as our reference point, giving us insights into the models' performance under normal, unaltered conditions.

The true essence of our research unfolds as we introduce adversarial attacks (described in Chapter 4) into the equation. These attacks are carefully designed to mimic real-world scenarios where video recognition systems may encounter subtle manipulations or deceptive inputs. We meticulously perform these attacks on the C3D, P3D, and Q3D models, aiming to understand how they respond when faced with adversarial challenges. We evaluate how well the models function both before and after these attacks are applied. These attacks take different forms, such as attempts to confuse the model or induce incorrect responses. At each stage of our experimentation, we scrutinize the accuracy of our models. This evaluation is pivotal in gauging the impact of adversarial attacks. Does the introduction of adversarial perturbations result in misclassifications or reduced accuracy? Or do our models demonstrate resilience in the face of these challenges? These questions guide our analysis as we delve into the results. We highlight specific observations, revealing how each model responds uniquely to different attack strategies.

The results of these experiments offer valuable insights into the areas where these models are susceptible and provide guidance on enhancing their capabilities to handle challenging scenarios effectively.

5.1 Model Performance's Evaluation: Measure of Accuracy

We will be using accuracy as a key parameter for judging model performance. This is not only common but also highly informative. In the subsequent sections, we will elaborate how accuracy was employed to evaluate model performance against various adversarial attacks and the valuable insights gained through this evaluation.

5.1.1 Using Accuracy as a Performance Metric

Accuracy is a widely used metric to evaluate the performance of machine learning models, particularly in classification tasks. It measures the ratio of correctly predicted instances to the total number of instances in the dataset. In the context of adversarial attacks on models, accuracy provides a clear and intuitive measure of how well the model is performing in the presence of adversarial manipulation i.e., ability to resist manipulation and make accurate predictions.

5.1.2 Calculation of Accuracy

To calculate accuracy, we constructed confusion matrices. [35] The confusion matrix (shown in Figure 5.1) provides a breakdown of model predictions, including true positives, true negatives, false positives, and false negatives. Accuracy is computed as shown in Equation 5.1.

Accuracy = $\frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$

Total number of predictions

Where, Number of correct predictions= True positives+ True negatives

Total number of predictions= True positives+ True negatives+ False positives + False negatives

Accuracy is mostly computed in percentage.

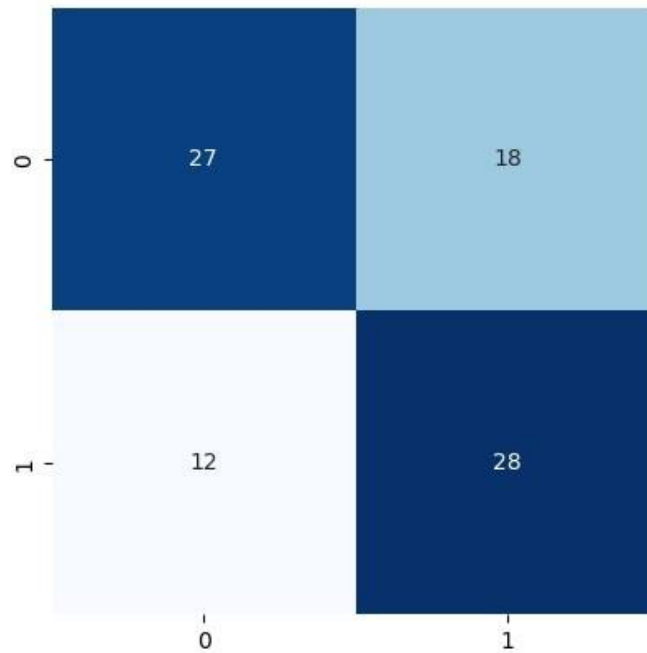


Figure 5.1 Confusion Matrix

From Figure 5.1, the Accuracy can be calculated as.

$$\text{Accuracy} = \frac{27+28}{27+28+18+12}$$

The total accuracy is calculated to be 0.64 which is 64%.

5.1.3 Insights Gained Through Accuracy Assessment in Adversarial Attacks

Assessing model performance against adversarial attacks using accuracy provides crucial insights. Initially, accuracy establishes a baseline on clean data, offering a reference point for assessing adversarial impact. A notable accuracy drop indicates vulnerability, while varying attack severity becomes evident by comparing accuracy across different adversarial tactics. Model robustness is quantified by comparing accuracy before and after attacks, with robust models maintaining reasonable accuracy levels. These insights inform the development of countermeasures, including adversarial training and defensive techniques, to bolster model resilience. In conclusion, accuracy proves invaluable for evaluating model performance, aiding the creation of more dependable AI systems capable of withstanding the challenges of an ever-evolving adversarial landscape.

5.2 Performance of the models on CrimesScene dataset before the attack

Our experiments have highlighted a standout model in terms of performance. The P3D model has shown outstanding capabilities by achieving remarkable results on the new test data. It boasts an impressive accuracy rate of 78%, firmly establishing itself as one of the top-performing models in our study. This demonstrates that P3D excels at making accurate predictions and recognizing things in videos, even when the videos are tricky or unfamiliar to the model.

While P3D performed exceptionally well, our experiment shows that C3D and Q3D, while competent, achieved a decent but comparatively lower accuracy rate of 64%. This difference encourages us to investigate further into the specific aspects of these models' performances, aiming to understand the factors that influenced their outcomes.

We've compiled all our experiment results in Table 5.1. It gives a quick and clear view of how accurate C3D, Q3D, and P3D were on the CrimeScene dataset.

Models	Accuracies
C3D	64%
P3D	78%
Q3D	64%

Table 5.1 Performance of Models before the attacks

The outcomes of these experiments extend beyond mere numbers. They provide insights into the strengths and limitations of these video recognition models, shedding light on their adaptability and resilience when faced with novel and challenging video content. These findings set the stage for a comprehensive analysis of the models' performance when subjected to various adversarial challenges, ultimately advancing our quest for more robust and secure video recognition systems.

5.3 Results of Adversarial Attacks on Crimes Scene Dataset

In the following sections, we conducted various adversarial attacks on our CrimesScene dataset, as outlined in Chapter 4. We evaluated the performance of our models when presented with these modified video frames. The subsequent sections provide insights into our models' performance across different attack scenarios.

5.3.1 Adversarial Attacks on Video Recognition Models using Adversarial Patch Technique

We conducted adversarial attacks on our video recognition models using the technique discussed in Section 4.1. and subsequently evaluated the outcomes.

5.3.1.1 Impact of Adversarial Patches in the Bottom right corner

We examined the effects of introducing three distinct adversarial patches into the bottom right corner of each frame within our video dataset. [\[22\]](#)

Following the inclusion of these patches, C3D's accuracy decreased by 21%, 14%, and 14% for Patch-1, Patch-2, and Patch-3, respectively. P3D, while initially robust with only a 3% accuracy drop, maintained its performance against all three patches, illustrating its resilience. Q3D, with an initial accuracy drop of 1.5%, also demonstrated adaptability to these adversarial patches.

5.3.1.2 Impact of Adversarial Patches in the Bottom left corner

Next, we strategically placed the three patches in the bottom left corner of each frame in our video dataset.

C3D exhibited resilience, with accuracy drops of 1.5%, 1.5%, and 7.75% against Patch-1, Patch-2, and Patch-3, respectively. P3D, initially showing an accuracy rise of 3.25%, maintained its accuracy against all three patches, reaffirming its robustness. Similarly, Q3D displayed adaptability, with accuracy drops of 1.5% and 1.5% against the first and second patches, and an increase of 4.75% against the third patch.

5.3.1.3 Impact of Adversarial Patches in the Top right corner

In this series of experiments, we strategically applied the three distinct adversarial patches to the top right corner of each frame in our video dataset.

C3D consistently exhibited an accuracy drop of 1.5% against all three patches, emphasizing its resilience to spatial perturbations in the top right corner. P3D maintained a consistent accuracy drop of 9.25% when subjected to all patches, reaffirming its robustness. Similarly, Q3D retained an accuracy drop of 1.5% in the presence of these adversarial manipulations.

5.3.1.4 Impact of Adversarial Patches in the Top left corner

Finally, we introduced three distinct adversarial patches into the top left corner of each frame in our video dataset.

C3D's accuracy drop, initially at 7.75%, showcased remarkable resilience against these patches, ultimately achieving an accuracy rise of 4.75% against all three. P3D also demonstrated unwavering performance, with an accuracy drop of 9.25% against each patch. Q3D, with its initial accuracy drop of 1.5%, maintained this level when subjected to adversarial manipulations. These findings shed light on the models' responses to spatial perturbations in various corners of the frames, emphasizing their distinct strengths and vulnerabilities in the face of adversarial attacks.

Table 5.2 shows the performance of models after adding the adversarial patches

Target Model	Patch in Corner	Accuracies		
		Patch-1	Patch-2	Patch-3
C3D	Bottom right	43%	50%	50%
	Bottom left	62.5%	62.5%	56.25%
	Top right	62.5%	62.5%	62.5%
	Top left	56.25%	68.75%	68.75%
P3D	Bottom right	75%	75%	75%
	Bottom left	81.25%	81.25%	81.25%
	Top right	68.75%	68.75%	68.75%
	Top left	68.75%	68.75%	68.75%
Q3D	Bottom right	62.5%	62.5%	62.5%
	Bottom left	62.5%	62.5%	62.5%
	Top right	62.5%	62.5%	62.5%
	Top left	62.5%	62.5%	62.5%

Table 5.2 Performance of models after adding the adversarial patches

These results illuminate the models' impressive resistance to adversarial attacks localized in the specific regions of video frames, emphasizing their robustness in this specific region. C3D employs a 3D convolutional approach, which may not be as robust when facing spatial and temporal perturbations introduced by adversarial attacks. In contrast, P3D and Q3D, with their pseudo-3D and quasi-3D architectures, respectively, may better capture temporal information while being less sensitive to frame-level alterations.

5.3.2 Adversarial Attacks on Video Recognition Models using Appending Adversarial Frame Technique

We conducted adversarial attacks on our video recognition models using the technique outlined in Section 4.2 and evaluated the results. Here are the key findings:

5.3.2.1 Impact of Appending Frames After 30 FPS on Video Recognition

Models

In these experiments, we introduced three distinct adversarial frames following a standard frame rate of 30 frames per second (FPS). [13] After the adversarial frames were incorporated at 30 FPS, C3D's accuracy experienced a decrease of 1.5% in Frame 1, followed by a further 7.75% decline in Frames 2 and 3. In contrast, P3D demonstrated greater resilience, with an accuracy reduction of 3% in Frame 1 and Frame 3, and a 9.25% drop in Frame 2. Q3D maintained a consistent accuracy decrease of 1.5% throughout the adversarial frames.

5.3.2.2 Impact of Appending Frames After 25 FPS on Video Recognition

Models

Moving on, we introduced three distinct adversarial frames after a standard frame rate of 25 FPS. As the adversarial frames were added at 25 FPS, C3D's accuracy decreased by 1.5% in Frame 1 and Frame 2, with a further 7.75% drop in Frame 3. P3D displayed a relatively robust performance, experiencing an accuracy drop of 3% in Frame 1, and a 9.25% reduction in both Frame 2 and Frame 3. Meanwhile, Q3D maintained a consistent accuracy decrease of 1.5% throughout the adversarial frames.

5.3.2.3 Impact of Appending Frames After 20 FPS on Video Recognition

Models

In the subsequent experiment, we added three distinct adversarial frames after a standard frame rate of 20 FPS. As the adversarial frames were introduced at 20 FPS, C3D's accuracy decreased by 1.5% in Frame 1 and Frame 2, followed by a 7.75% reduction in Frame 3. P3D continued to demonstrate resilience, with a 3% drop in accuracy in Frame 1 and a 9.25% decrease in both Frame 2 and Frame 3. Q3D maintained a consistent accuracy reduction of 1.5% across all the adversarial frames.

5.3.2.4 Impact of Appending Frames After 15 FPS on Video Recognition

Models

In the next experiment, three distinct adversarial frames were inserted after a standard frame rate of 15 FPS. As the adversarial frames were integrated, C3D's accuracy

dropped by 1.5% in Frame 1 and Frame 2, with a further 7.75% decline in Frame 3. In contrast, P3D displayed varying performance, with a 9.25% reduction in accuracy in Frame 1 and Frame 3, while experiencing only a 3% drop in Frame 2. Q3D saw a drop in accuracy of 7.75% in Frame 2 and Frame 3, while maintaining a consistent 1.5% drop in accuracy in Frame 1.

Table 5.3 shows the performance of models with appended adversarial frames beyond a certain FPS.

Target Model	FPS	Accuracies		
		Frame-1	Frame-2	Frame-3
C3D	30	62.5%	56.25%	56.25%
	25	62.5%	62.5%	56.25%
	20	62.5%	62.5%	56.25%
	15	62.5%	62.5%	56.25%
P3D	30	75%	68.75%	75%
	25	75%	68.75%	68.75%
	20	68.75%	68.75%	68.75%
	15	68.75%	75%	68.75%
Q3D	30	62.5%	62.5%	62.5%
	25	62.5%	62.5%	62.5%
	20	62.5%	62.5%	62.5%
	15	62.5%	56.25%	56.25%

Table 5.3 performance of models with appended adversarial frames beyond a certain FPS

These findings shed light on P3D's vulnerability in this context. P3D's use of pseudo-3D convolutional filters to capture motion information might make it more sensitive to adversarial perturbations in the frames, as these perturbations can affect the perception of motion cues.

5.3.3 Adversarial Attacks on Video Recognition Models through Gaussian Noise

We conducted adversarial attacks on our video recognition models using the technique outlined in Section 4.3 and evaluated the results. [26] Notably, the range of values for μ was systematically varied between -5 and 5, and σ within the range of 1 to 5. Here are the key findings:

5.3.3.1 Adversarial Attacks with Low Gaussian Noise Parameters

In the first series of experiments, we employed low values for both μ and σ , ranging from -5 to 0 for μ and 1 to 2.25 for σ . These attacks revealed distinct vulnerabilities within the models. C3D experienced a significant accuracy drop of 14%, indicating its sensitivity to this specific form of noise. P3D followed suit with a reduction in accuracy, while Q3D consistently exhibited a 14% accuracy drop, underscoring its limited adaptability to low μ and σ values in Gaussian noise attacks.

5.3.3.2 Adversarial Attacks Using Gaussian Noise: Low Mu and High Sigma Values

In the subsequent set of experiments, we introduced Gaussian noise with low μ and higher σ values. Surprisingly, C3D showcased an interesting pattern, with its accuracy gradually improving as σ increased from 3.75 to 4.75, stabilizing at a mere 1.5% drop. Conversely, P3D displayed fluctuations in accuracy, hitting its lowest point with a 9.25% drop when σ was set to 4.75. Meanwhile, Q3D remained relatively stable, maintaining a 14% accuracy drop.

5.3.3.3 Adversarial Attacks Using Gaussian Noise: High Mu and Low Sigma Values

Moving forward, we explored the impact of high μ and low σ values on the models. Here, C3D's accuracy drop remained consistent at 1.5%, showcasing its resilience to this specific noise configuration. P3D demonstrated intriguing behavior, achieving increased accuracy drops, notably at μ values of 4 and 5, where it experienced a rise in accuracy drop to 3.25%. Q3D also displayed an accuracy increase under these conditions, particularly at μ values of 4 and 5.

5.3.3.4 Adversarial Attacks with High Gaussian Noise Parameters

Lastly, we delved into adversarial attacks with high values for both μ and σ . In this scenario, C3D maintained a 1.5% accuracy drop, highlighting its robustness against this noise type. P3D exhibited fluctuations, reaching notable accuracy drops at μ values of 1 and 3, with drops of 7.75% and 3%, respectively. Q3D displayed a minimal accuracy decrease at $\mu = 5$ and $\sigma = 5$, reducing by 7.25% from the original 14%.

These findings underscore the intricate relationships between model architectures and the nuanced parameters of Gaussian noise, shedding light on the models' diverse responses under varying noise conditions.

Table 5.4 shows the performance of the models after adding Gaussian noise with varying parameters.

Target Model	Low μ & σ		Accuracies
	μ	σ	
C3D	-5	1	50%
	-4	1.25	50%
	-3	1.5	50%
	-2	1.75	62.5%
	-1	2	62.5%
	0	2.25	68.75%
P3D	-5	1	50%
	-4	1.25	50%
	-3	1.5	50%
	-2	1.75	56.25%
	-1	2	56.25%
	0	2.25	68.75%
Q3D	-5	1	50%
	-4	1.25	50%
	-3	1.5	50%
	-2	1.75	50%
	-1	2	50%

CHAPTER 5: EXPERIMENTS & RESULTS

	0	2.25	50%
Low μ & High σ			
	μ	σ	
C3D	-5	3.75	50%
	-4	4	56.25%
	-3	4.25	62.5%
	-2	4.5	62.5%
	-1	4.75	62.5%
	0	5	62.5%
	P3D	-5	3.75
-4		4	50%
-3		4.25	50%
-2		4.5	50%
-1		4.75	68.75%
0		5	62.5%
Q3D		-5	3.75
	-4	4	50%
	-3	4.25	50%
	-2	4.5	50%
	-1	4.75	50%
	0	5	50%
	High μ & Low σ		
	μ	σ	
C3D	0	1	62.5%
	1	1.25	62.5%
	2	1.5	62.5%
	3	1.75	62.5%
	4	2	62.5%
	5	2.25	62.5%
	P3D	0	1
1		1.25	75%
2		1.5	75%
3		1.75	75%

CHAPTER 5: EXPERIMENTS & RESULTS

	4	2	75%
	5	2.25	75%
Q3D	0	1	50%
	1	1.25	56.25%
	2	1.5	62.5%
	3	1.75	62.5%
	4	2	62.5%
	5	2.25	62.5%
High μ & σ			
	μ	σ	
C3D	0	3.75	62.5%
	1	4	68.75%
	2	4.25	62.5%
	3	4.5	62.5%
	4	4.75	62.5%
	5	5	62.5%
P3D	0	3.75	75%
	1	4	68.75%
	2	4.25	68.75%
	3	4.5	75%
	4	4.75	75%
	5	5	75%
Q3D	0	3.75	50%
	1	4	50%
	2	4.25	50%
	3	4.5	50%
	4	4.75	50%
	5	5	56.25%

Table 5.4 performance of the models after adding Gaussian noise with varying parameters.

5.3.4 Adversarial Attacks on Video Recognition Models Through Contrast Adjustment

We conducted adversarial attacks on our video recognition models using the technique outlined in Section 4.4 and evaluated the results. It's noteworthy that the range of values for alpha was systematically adjusted between 0.1 to 3, and beta within the range of -255 to 255. Here are the key findings:

5.3.4.1 Low values of alpha (α) and beta (β)

In this study, [28] we conducted adversarial attacks on video recognition models by applying contrast adjustments to the frames of our video dataset. The table above summarizes the impact of these attacks on three different models: C3D, P3D, and Q3D. We employed low values of alpha (α) and beta (β) to introduce contrast adjustments. The results reveal that as we increased the values of α and β , the models' accuracy underwent varying degrees of change.

As we introduced contrast adjustments with α ranging from 0.01 to 0.75 and β decreasing from -255 to -50, C3D's accuracy consistently decreased, ultimately dropping by 7.75%. This suggests that C3D is relatively robust against low-level contrast adjustments but becomes increasingly susceptible to more significant changes.

Similarly, the P3D model also showed a pattern of decreasing accuracy as we applied contrast adjustments. The accuracy of P3D dropped by 21.75% with the highest values of α and β , indicating its vulnerability to these adversarial manipulations.

Q3D also experienced accuracy reductions as contrast adjustments were introduced. The most substantial drop occurred when α was 0.75 and β was -50, leading to a final accuracy drop of 1.5%. Q3D, therefore, demonstrated a sensitivity to contrast alterations but exhibited a more stable performance compared to P3D.

5.3.4.2 Low values of alpha (α) and High values of beta (β)

Next, we applied contrast adjustments to the frames of our video dataset, using low values of alpha (α) and high values of beta (β). The table above summarizes the impact of these attacks on three different models: C3D, P3D, and Q3D. As we increased the

values of α and set β to high positive values, the models' accuracy exhibited varying degrees of change.

As we introduced contrast adjustments with increasing α and high β values, C3D's accuracy drop remained relatively stable at 14%. This suggests that C3D is relatively resilient against these specific contrast alterations.

Conversely, the P3D model demonstrated a pattern of accuracy improvement with the introduction of higher α and β values. Its accuracy increased to 68.75% in the most extreme case, indicating that P3D could benefit from these contrast adjustments when the values are sufficiently high.

Q3D remained largely unaffected by the contrast adjustments, with its accuracy drop consistently at 14% regardless of the values of α and high β . This suggests that Q3D is more robust to these particular adversarial manipulations.

5.3.4.3 High values of alpha (α) and Low values of beta (β)

Next, we conducted adversarial attacks on video recognition models by applying contrast adjustments to the frames of our video dataset, using high values of alpha (α) and low values of beta (β). The table above summarizes the impact of these attacks on three distinct models: C3D, P3D, and Q3D. As α was consistently set at 1.0 and we decreased β from -255 to -50, the models' accuracy exhibited varying degrees of change.

As contrast adjustments were introduced with progressively lower values of β , the C3D model's accuracy showed fluctuations, ranging from an accuracy drop of 26.5% to a high of 1.5%. This indicates that C3D is susceptible to these particular contrast manipulations when high α values are applied.

P3D displayed a range of responses to the contrast adjustments with its accuracy drop fluctuating from 21.75% to 3%, with the highest accuracy achieved at $\alpha = 2.0$ and $\beta = -150$. This suggests that P3D might exhibit some resistance to these specific adversarial perturbations under certain conditions when high α values are used.

Likewise, Q3D demonstrated a spectrum of responses to the attacks, resulting in fluctuations in accuracy ranging from a 14% decline to a notable 4.75% increase.

Notably, the most substantial accuracy increase occurred at $\alpha = 1.5$ and $\beta = -200$. This indicates that Q3D's performance is subject to fluctuations and may even benefit from the adversarial manipulations under specific conditions with high α values and low β values.

5.3.4.4 High values of alpha (α) and beta (β)

Next, we investigated the effects of adversarial attacks on video recognition models by applying contrast adjustments to the frames of our video dataset, using high values of both alpha (α) and beta (β). The table above summarizes the outcomes for three distinct models: C3D, P3D, and Q3D. With a consistent α value of 1.0 and increasing β values from 50 to 255, the models' accuracy demonstrated diverse changes.

As contrast adjustments were introduced with higher β values, C3D's accuracy showed fluctuations but remained relatively stable, with the lowest accuracy drop recorded at 14% and the highest at 1.5%. This suggests that C3D is somewhat resilient to these specific contrast manipulations when high α and β values are applied.

P3D displayed a decline in accuracy as we increased β from 50 to 255. Its accuracy drop ranged from 28% to 3%, with the highest drop occurring at $\alpha = 1.0$ and $\beta = 50$. This indicates that P3D can be sensitive to such adversarial perturbations when high α and β values are utilized.

Q3D remained mostly unaffected by the contrast adjustments, with its accuracy drop consistently at 14% regardless of the values of α and high β . This suggests that Q3D is quite robust to these specific adversarial manipulations.

Table 5.5 shows the performance of the models after changing the contrast by varying different parameters.

Target Model	Low α & β		Accuracies
	α	β	
C3D	0.01	-255	50%
	0.1	-200	50%
	0.25	-150	50%

CHAPTER 5: EXPERIMENTS & RESULTS

	0.5	-100	56.25%
	0.75	-50	56.25%
P3D	0.01	-255	50%
	0.1	-200	50%
	0.25	-150	50%
	0.5	-100	56.25%
	0.75	-50	75%
Q3D	0.01	-255	50%
	0.1	-200	50%
	0.25	-150	50%
	0.5	-100	50%
	0.75	-50	62.5%
Low α & High β			
	α	β	
C3D	0.01	255	50%
	0.1	200	50%
	0.25	150	62.5%
	0.5	100	62.5%
	0.75	50	62.5%
P3D	0.01	255	50%
	0.1	200	50%
	0.25	150	50%
	0.5	100	68.75%
	0.75	50	68.75%
Q3D	0.01	255	50%
	0.1	200	50%
	0.25	150	50%
	0.5	100	50%
	0.75	50	50%
High α & Low β			
	α	β	
C3D	1.0	-255	37.5%
	1.5	-200	62.5%

CHAPTER 5: EXPERIMENTS & RESULTS

	2.0	-150	43.75%
	2.5	-100	62.5%
	3.0	-50	37.5%
P3D	1.0	-255	56.25%
	1.5	-200	56.25%
	2.0	-150	75%
	2.5	-100	68.75%
	3.0	-50	56.25%
Q3D	1.0	-255	50%
	1.5	-200	68.75%
	2.0	-150	68.75%
	2.5	-100	56.25%
	3.0	-50	56.25%
High α & β			
	α	β	
C3D	1.0	255	62.5%
	1.5	200	62.5%
	2.0	150	56.25%
	2.5	100	50%
	3.0	50	50%
P3D	1.0	255	75%
	1.5	200	56.25%
	2.0	150	56.25%
	2.5	100	50%
	3.0	50	50%
Q3D	1.0	255	50%
	1.5	200	50%
	2.0	150	50%
	2.5	100	50%
	3.0	50	50%

Table 5.5 Performance of the models after changing the contrast by varying different parameters.

In these experiments, various contrast adjustment-based adversarial attacks were applied to video recognition models, exploring four scenarios based on alpha (α) and beta (β) values. The models exhibited diverse responses: low α and β led to susceptibility in C3D, P3D's vulnerability, and Q3D's relative stability. Conversely, low α and high β values showcased C3D's resilience, P3D's occasional benefit, and Q3D's robustness. High α and low β values introduced complexity in C3D, varying sensitivity and resistance in P3D, and fluctuations and occasional benefit in Q3D. High α and β values highlighted C3D's relative resilience, P3D's sensitivity to high α , and Q3D's robustness. These findings underscore the importance of understanding model-specific behaviors for enhancing security and robustness in video recognition tasks.

5.3.5 Adversarial Attacks on Video Recognition Models Through Salt & Pepper Noise

We conducted adversarial attacks on our video recognition models using the technique outlined in Section 4.5 and evaluated the results. Here are the key findings:

5.3.5.1 Uniform Salt and Pepper Noise Probability in Adversarial Attacks

In this study, we conducted adversarial attacks on video recognition models by introducing visual distortions to the frames of our video dataset, using the same values of salt and pepper noise probability (salt_prob and pepper_prob). [29] The table above summarizes the impact of these attacks on three distinct models: C3D, P3D, and Q3D.

When introducing visual distortions with salt_prob and pepper_prob set at 0.01, C3D exhibited a moderate accuracy drop of 7.75%, which improved to a 1.5% decrease as salt and pepper noise probability rose to 0.05, indicating some adaptability to these conditions. However, C3D's accuracy started to drop as salt_prob and pepper_prob reached 0.1 and beyond, ultimately stabilizing at a drop of 14%, suggesting increased susceptibility to motion blur.

P3D, on the other hand, exhibited a different pattern, it started with a minimal 3% accuracy decrease and maintained this level when salt_prob and pepper_prob were low. However, as these probabilities increased, P3D's accuracy intensified, reaching 14%, implying its vulnerability with higher salt and pepper noise probabilities.

Q3D showed a notable accuracy drop as the salt and pepper noise probability increased, ultimately stabilizing at 14% decrease. This suggests Q3D's sensitivity to motion blur under these conditions.

5.3.5.2 Varied Salt and Pepper Noise Effects in Adversarial Attacks

In our subsequent experiments, we conducted adversarial attacks on video recognition models by introducing a range of perturbations to the frames of our video dataset. Notably, we employed varying combinations of salt and pepper noise probabilities, with high values for salt_prob and low values for pepper_prob in some cases, and vice versa. The table above summarizes the outcomes for three distinct models: C3D, P3D, and Q3D.

When we introduced these perturbations with high salt_prob (ranging from 0.5 to 1) and low pepper_prob (ranging from 0.01 to 0.1), C3D's accuracy exhibited fluctuations, ultimately revealing a 4.75% increase. This suggests that C3D displayed some resilience to these specific conditions under varying salt and pepper noise probabilities.

Conversely, P3D's accuracy experienced a marginal drop of 3% under the conditions of high salt and low pepper noise probabilities. However, its accuracy showed variability when these probabilities were reversed, indicating vulnerability to specific combinations of perturbations.

Q3D remained largely unaffected by the changing salt and pepper noise probabilities, with its accuracy drop consistently maintained at 14%. This suggests that Q3D exhibited robustness against the introduced perturbations under these varying conditions.

In our study on adversarial attacks, we investigated the impact of visual distortions, specifically salt and pepper noise, on three video recognition models: C3D, P3D, and Q3D. When using uniform salt and pepper noise probabilities, C3D displayed adaptability to low noise levels but increased susceptibility as noise probabilities rose. P3D showed vulnerability with higher noise probabilities, resulting in a significant accuracy drop. In varied noise conditions, C3D demonstrated resilience to specific combinations, while P3D exhibited sensitivity and variability. Q3D remained robust

and unaffected by changing noise probabilities. This highlights the varying responses of these models to adversarial attacks.

Table 5.6 shows the performance of the models after adding Salt and Pepper noise with varying parameters.

Target Model	Uniform salt & pepper noise		Accuracies
	salt_prob	pepper_prob	
C3D	0.01	0.01	56.25%
	0.05	0.05	62.5%
	0.1	0.1	68.75%
	0.25	0.25	56.25%
	0.5	0.5	50%
	0.75	0.75	50%
P3D	0.01	0.01	75%
	0.05	0.05	75%
	0.1	0.1	68.75%
	0.25	0.25	50%
	0.5	0.5	50%
	0.75	0.75	50%
Q3D	0.01	0.01	62.5%
	0.05	0.05	56.25%
	0.1	0.1	50%
	0.25	0.25	50%
	0.5	0.5	50%
	0.75	0.75	50%
Variable salt and pepper noise			
	salt_prob	pepper_prob	
C3D	0.5	0.01	56.25%
	0.75	0.05	50%
	1	0.1	50%
	0.01	0.5	62.5%

	0.5	0.75	62.5%
	0.1	1	68.75%
P3D	0.5	0.01	50%
	0.75	0.05	50%
	1	0.1	50%
	0.01	0.5	75%
	0.5	0.75	56.25%
	0.1	1	56.25%
Q3D	0.5	0.01	50%
	0.75	0.05	50%
	1	0.1	50%
	0.01	0.5	50%
	0.5	0.75	50%
	0.1	1	50%

Table 5.6 performance of the models after adding Salt and Pepper noise with varying parameters.

5.3.6 Adversarial Attacks on Video Recognition Models using Motion Blur Technique

We conducted adversarial attacks on our video recognition models using the technique outlined in Section 4.6 and evaluated the results. Here are the key findings:

5.3.6.1 Adversarial Attacks with Horizontal Motion Blur and Varying Kernel Sizes

In our study, we conducted adversarial attacks on video recognition models by introducing motion blur to the frames of our video dataset, while keeping the orientation consistently horizontal. [30] We varied the kernel size, ranging from 5 to 30, to evaluate the impact on three distinct models: C3D, P3D, and Q3D. The results, as presented in the table, indicate that these models exhibited varying levels of sensitivity to motion blur under these conditions.

C3D's accuracy consistently decreased across all kernel sizes, with the greatest impact observed at a 5x5 kernel size, resulting in a substantial 20% drop from its original

accuracy. P3D also showed a decrease in accuracy, although it was less pronounced, with a 10% drop when the kernel size was set to 5x5. In contrast, Q3D exhibited a mixed response, with some kernel sizes leading to increased accuracy, possibly due to the model's robustness against certain motion blur conditions.

5.3.6.2 Adversarial Attacks with Vertical Motion Blur and Varying Kernel Sizes

Our exploration continued with adversarial attacks that maintained a constant vertical orientation. We systematically altered the kernel size, ranging from 5 to 30, to assess the models' reactions to vertical motion blur.

C3D's accuracy remained relatively stable across different kernel sizes, indicating a consistent performance even in the presence of motion blur. P3D, on the other hand, exhibited mixed results, with some kernel sizes causing a drop in accuracy, particularly at 15x15 and 20x20 kernel sizes. Q3D displayed a similar pattern, with certain kernel sizes leading to decreased accuracy, most notably at 15x15.

5.3.6.3 Adversarial Attacks with Diagonal (45°) Motion Blur and Varying Kernel Sizes

In the subsequent phase, we introduced diagonal motion blur at a 45-degree angle, with varying kernel sizes ranging from 5 to 30, as part of our adversarial attacks. The models' performance, as reflected in the accuracies, shed light on their reactions to diagonal motion blur in these conditions.

C3D consistently demonstrated a relatively stable accuracy across different kernel sizes, maintaining its performance in the presence of diagonal motion blur. P3D, in contrast, displayed consistently high accuracy levels, with occasional fluctuations, especially at larger kernel sizes where its accuracy increased by 3.25%. Q3D, on the other hand, showed sensitivity to diagonal motion blur, with accuracy decreasing as the kernel size increased.

5.3.6.4 Adversarial Attacks with Diagonal (135°) Motion Blur and Varying Kernel Sizes

Moving on, we conducted adversarial attacks by applying diagonal motion blur at a 135-degree angle to the frames of our video dataset, altering the kernel sizes from 5 to 30. This evaluation aimed to understand how these models respond to diagonal motion blur at this specific orientation and with different kernel sizes.

C3D maintained relatively consistent accuracy across different kernel sizes, exhibiting a stable performance under the influence of diagonal motion blur at a 135-degree angle. In contrast, P3D showed a remarkable ability to resist this type of perturbation, with high accuracy levels remaining largely unchanged, even at larger kernel sizes where its accuracy increased by 3.25%. Q3D, however, displayed sensitivity to diagonal motion blur at 135 degrees, with a consistent decrease in accuracy as the kernel size increased.

5.3.6.5 Adversarial Attacks with Cross Motion Blur and Varying Kernel Sizes

In our latest series of experiments, we conducted adversarial attacks by introducing motion blur to the frames of our video dataset with a unique twist—maintaining a "cross" orientation. This involved deliberately setting the direction of motion blur perpendicular to the primary orientation of the objects or scenes in the videos. The table above summarizes the outcomes for three distinct models: C3D, P3D, and Q3D.

For C3D, the introduction of cross orientation motion blur resulted in a 6.25% increase in accuracy at a kernel size of 20, indicating an interesting resilience under these specific conditions. P3D, on the other hand, showed variability in accuracy, with some fluctuations in response to the cross-orientation motion blur. Nevertheless, it remained relatively stable at higher kernel sizes, indicating a moderate level of robustness. Q3D's accuracy remained mostly unaffected under these conditions, suggesting a degree of stability when confronted with cross orientation motion blur.

In this study of adversarial attacks on video recognition models, we explored their responses to different types of motion blur and varied kernel sizes. C3D showed varied sensitivity across motion blur types, with pronounced accuracy drops in horizontal and

diagonal motion blur scenarios. P3D displayed a degree of robustness, maintaining high accuracy under certain conditions. Q3D exhibited mixed responses to the perturbations, indicating its sensitivity to particular motion blur orientations. These findings highlight the nuanced behavior of these models under diverse motion blur conditions and kernel sizes.

Table 5.7 shows the performance of the model after the introduction of a motion blur filter in video frames with diverse parameters.

Target Model	Horizontal Orientation	Accuracies
	Kernel Size	
C3D	5	50%
	10	50%
	15	50%
	20	62.5%
	25	62.5%
	30	68.75%
P3D	5	50%
	10	50%
	15	50%
	20	56.25%
	25	56.25%
	30	68.75%
Q3D	5	50%
	10	50%
	15	50%
	20	50%
	25	50%
	30	50%
Vertical Orientation		
Kernel size		
C3D	5	50%
	10	56.25%
	15	62.5%
	20	62.5%
	25	62.5%
	30	62.5%
P3D	5	50%
	10	50%

CHAPTER 5: EXPERIMENTS & RESULTS

	15	50%
	20	50%
	25	68.75%
	30	62.5%
Q3D	5	50%
	10	50%
	15	50%
	20	50%
	25	50%
	30	50%
Diagonal Orientation		
Kernel size		
C3D	5	62.5%
	10	62.5%
	15	62.5%
	20	62.5%
	25	62.5%
	30	62.5%
P3D	5	75%
	10	75%
	15	75%
	20	75%
	25	75%
	30	75%
Q3D	5	50%
	10	56.25%
	15	62.5%
	20	62.5%
	25	62.5%
	30	62.5%
Diagonal (135°) Orientation		
Kernel size		
C3D	5	62.5%

CHAPTER 5: EXPERIMENTS & RESULTS

	10	68.75%
	15	62.5%
	20	62.5%
	25	62.5%
	30	62.5%
P3D	5	75%
	10	68.75%
	15	68.75%
	20	75%
	25	75%
	30	75%
Q3D	5	50%
	10	50%
	15	50%
	20	50%
	25	50%
	30	56.25%
Cross Orientation		
Kernel size		
C3D	5	43.75%
	10	43.75%
	15	43.75%
	20	50%
	25	50%
	30	50%
P3D	5	62.5%
	10	68.75%
	15	62.5%
	20	62.5%
	25	62.5%
	30	62.5%
Q3D	5	56.25%
	10	50%

15	50%
20	50%
25	50%
30	50%

Table 5.7 Performance of the model after the introduction of a motion blur filter in video frames with diverse parameters.

5.3.7 Adversarial Attacks on Video Recognition Models using Frame Dropping Technique

We conducted adversarial attacks on our video recognition models using the technique outlined in Section 4.7 and evaluated the results. Here are the key findings:

After subjecting video recognition models to adversarial attacks using the frame dropping technique with different frame drop rates [31], a notable impact on their accuracies has been observed. The effectiveness of the frame dropping attack can be seen in the significant reductions in accuracy across the models. For the C3D model, a frame drop rate of 2 resulted in a decrease in accuracy by 1.5%, while the accuracy of P3D at the same frame drop rate was reduced by 28%. A frame drop rate of 5 led to variable outcomes with C3D and Q3D but resulted in an accuracy drop of 9.75% for the P3D model.

Frame dropping can considerably hinder the models' abilities to correctly classify actions and objects in videos, highlighting the importance of developing robust defense mechanisms against such adversarial attacks. This reduction in accuracy underscores the need for continuous research and improvement to bolster the security and reliability of video recognition systems in the face of adversarial threats.

Table 5.8 shows the performance of the models after dropping a certain number of frames.

Target Model	Frame Drop Rate	Accuracies
C3D	2	62.5%
	3	56.25%
	5	62.5%
P3D	2	50%
	3	50%
	5	68.25%
Q3D	2	50%
	3	50%
	5	62.5%

Table 5.8 performance of the models after dropping a certain number of frames.

5.4 Comparison of Model Performance in Handling Additional Features Related to Learned Patterns

5.4.1 Experimental Setup:

Both our models (C3D, P3D, Q3D) and another model (C3D, LRCN) underwent an experiment where extra features, specifically tied to learned patterns, were introduced to the video data.

5.4.2 Impact on Our Models:

Upon introducing these features, our models experienced a noticeable decline in accuracy. This drop was attributed to an increased focus on certain patch-related features, learned during training but not directly relevant to the video analysis task.

5.4.3 Behavior of the Other Model:

In contrast, the other model demonstrated a more resilient response to extraneous features. The impact on accuracy was either less pronounced or negligible compared to

our models. This suggests that the other model may possess a more robust mechanism for handling non-task-specific features.

5.4.4 Specifics in the Case of C3D:

For the C3D model, commonly used in video analysis, our model exhibited a disproportionate emphasis on specific patch-related features, leading to a maximum 21% accuracy drop upon feature addition. The other model maintained a more balanced consideration of features, preserving higher accuracy despite the introduction of non-task-related information.

5.4.5 Implications for Our Model:

Observing our model's tendency to overly prioritize certain features, especially patch-related ones, highlights challenges in its adaptability to diverse video scenarios. This overemphasis may result in suboptimal performance when faced with videos containing features not directly aligned with learned patterns during training.

In conclusion, the comparative analysis between our model and another model underscores the importance of a model's robustness in handling extraneous features. While the other model exhibited a more stable performance in the face of introduced patterns, our model demonstrated sensitivity, particularly in the case of the C3D model. These insights inform potential avenues for refining our model to enhance its adaptability and effectiveness in diverse video analysis scenarios.

5.5 Defenses Against Adversarial Attacks on Video Recognition Models

Within the realm of video recognition models, I engaged in a study to understand their response to challenges. Initially, I tested these models deliberately, introducing changes to the input data. This process allowed me to identify areas where the models might struggle or make errors.

Subsequently, the focus shifted to fortifying these models. I developed strategic defenses to shield them from being misled by intentional confusion. These defenses act

as safeguards, ensuring that the models can maintain accurate video recognition even when faced with attempts to cause confusion.

This two-phase approach—first, identifying vulnerabilities, and then constructing defenses—aims to improve the reliability and security of video recognition systems in the dynamic field of artificial intelligence.

Some of the major defenses used during this research are:

5.5.1 Median Filtering

Median filtering is a robust defense strategy against salt and pepper noise, a type of disruptive noise in images and videos. This technique works by replacing each pixel with the median value within its local neighborhood, effectively eliminating isolated extreme values introduced by salt and pepper noise. Unlike other filtering methods, median filtering maintains the structural details of the image while effectively reducing noise. [32] This makes it especially suitable for preserving the integrity of video frames in the presence of sporadic and unpredictable noise. The simplicity and efficiency of median filtering contribute to its widespread use as an effective defense mechanism in video recognition models, ensuring improved accuracy and reliability by mitigating the impact of salt and pepper noise.

5.5.2 Gaussian Blur

Gaussian blur stands as a practical defense mechanism against Gaussian noise, a type of random noise that manifests as a smooth, continuous variation in pixel intensity. When applied as a defense, Gaussian blur mitigates the disruptive effects of Gaussian noise by averaging pixel values within a local neighborhood, effectively smoothing out irregularities in intensity. This process involves convolving the image or video frame with a Gaussian kernel, resulting in a blurring effect that reduces the impact of high-frequency variations introduced by Gaussian noise. The strength of the blur is determined by the standard deviation of the Gaussian kernel, allowing for a customizable level of noise reduction. [33] This defense strategy is particularly effective in scenarios where maintaining a visually coherent appearance is essential, as

it targets the gradual fluctuations characteristic of Gaussian noise without sacrificing overall image or video quality.

5.5.3 Deblurring Filter

Deblurring serves as a potent defense mechanism against motion blur attacks, where intentional blurring is introduced to compromise image or video recognition. In response to such attacks, deblurring algorithms are employed to reverse or minimize the effects of motion blur. These algorithms analyze the characteristics of the blur and apply an inverse operation to restore sharpness and clarity to distorted regions. This defense strategy proves effective in countering intentional blurring introduced by adversaries, enhancing the accuracy of recognition models by ensuring a clearer representation of the original content. [34] Deblurring is particularly valuable in scenarios where maintaining visual fidelity is paramount, offering a reliable means to mitigate the intentional blurring introduced to undermine the performance of recognition models.

5.5.4 Experiments & Results

During our defense experiments, we implemented protective measures against adversarial attacks on video recognition models. We employed diverse defense strategies, including Motion Blur defense, Median Blur defense, and Gaussian Noise. The table above outlines the effectiveness of these defenses across three distinct models: C3D, P3D, and Q3D.

For C3D, both the Motion Blur and Median Blur defenses consistently exhibited a 56.25% accuracy improvement, showcasing the model's heightened resilience. Additionally, the Gaussian Noise defense resulted in a 50% accuracy improvement, further reinforcing C3D's robustness against adversarial attacks.

P3D displayed varying degrees of success with the applied defenses, with the Motion Blur defense yielding an impressive 81.25% accuracy improvement, emphasizing its notable resistance. The Median Blur defense achieved a 68.75% improvement, while the Gaussian Noise defense contributed a 50% accuracy boost, indicating P3D's adaptability to specific defense strategies, particularly Motion Blur.

Q3D, the third model under consideration, consistently demonstrated a 56.25% accuracy improvement with the Motion Blur defense, a 62.5% improvement with the Median Blur defense, and a 50% improvement with the Gaussian Noise defense. These defense mechanisms collectively underscored Q3D's enhanced resistance to adversarial attacks, highlighting its robust performance across diverse defense strategies.

Table 5.9 shows the performance of our models against these defense strategies.

Model	Deblurring Filter	Median Filtering	Gaussian Blur
C3D	56.25%	56.25%	50%
	56.25%	62.5%	62.5%
	56.25%	50%	62.5%
	56.25%	50%	62.5%
	56.25%	50%	62.5%
	56.25%	50%	62.5%
P3D	81.25%	68.75%	50%
	68.75%	75%	68.75%
	75%	50%	81.25%
	62.5%	50%	81.25%
	62.5%	50%	68.75%
	62.5%	56.25%	75%
Q3D	56.25%	62.5%	50%
	56.25%	50%	50%
	50%	50%	50%
	56.25%	50%	50%
	50%	50%	50%
	56.25%	50%	50%

Table 5.9 Performance of our models against these defense strategies

In summary, our application of Motion Blur, Median Blur, and Gaussian Noise defenses yielded positive results across all three video recognition models. These defenses significantly contributed to fortifying the models, showcasing increased accuracy and resilience against potential adversarial perturbations.

Chapter 6

Conclusion & Future Work

In conclusion, this thesis has undertaken a comprehensive investigation into the vulnerability of video recognition models, specifically C3D, P3D, and Q3D, to adversarial attacks using the Crimes Scene dataset. The empirical analysis revealed intricate patterns of susceptibility, with some attacks consistently diminishing model accuracy, while others induced a constant impact. Comparative assessments against other models in terms of accuracy provided valuable insights into the relative performance of the three models within the domain.

Moreover, the study did not merely stop at identifying vulnerabilities; it extended to the development and evaluation of defensive strategies aimed at fortifying the models against adversarial threats. The implementation of these defense mechanisms showcased promising results in enhancing the resilience of the video recognition models, marking a crucial step towards addressing the security challenges posed by adversarial attacks.

The findings of this research emphasize the nuanced nature of model vulnerabilities and the need for tailored defensive strategies. As the field of adversarial attacks continues to evolve, the insights gained from this study serve as a foundation for further research and development in the realm of video recognition model security.

While this thesis has made significant strides in understanding and mitigating adversarial attacks on video recognition models, there remains a plethora of avenues for future exploration. Firstly, expanding the scope of the study to include diverse datasets and real-world scenarios would provide a more comprehensive understanding of model behavior and performance in practical applications.

Additionally, exploring novel adversarial attack techniques and adapting defensive strategies to address emerging threats would contribute to the ongoing evolution of model security. Investigating the transferability of defenses across different model architectures and datasets could offer insights into the generalizability of protective measures.

Furthermore, the incorporation of explain ability and interpretability techniques into defensive strategies would enhance our ability to comprehend the decision-making processes of models under adversarial conditions. This would not only bolster the robustness of the models but also foster trust and transparency in their applications.

In conclusion, future research endeavors should aim to deepen our understanding of adversarial threats in video recognition models, refine defensive mechanisms, and explore the broader implications of these findings in real-world scenarios. The pursuit of these avenues promises to contribute significantly to the ongoing efforts to secure video recognition models against adversarial challenges.

Bibliography

- [1] Athalye, A., Carlini, N., & Wagner, D. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples.
- [2] A. Kurakin, I. J. Goodfellow, and S. Bengio, "ADVERSARIAL MACHINE LEARNING AT SCALE," in Proc. International Conference on Learning Representations (ICLR), <https://arxiv.org/abs/1611.01236>, 2017.
- [3] Y. Xu, X. Liu, M. Yin, T. Hu, and K. Ding, "SPARSE ADVERSARIAL ATTACK FOR VIDEO VIA GRADIENT-BASED KEYFRAME SELECTION," in Proc. 2022 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2022, doi: 10.1109/ICASSP43922.2022.9747698.
- [4] H. Heo, D. Ko, J. Lee, Y. Hong and H. J. Kim, "Search-and-Attack: Temporally Sparse Adversarial Perturbations on Videos," in IEEE Access, vol. 9, pp. 146938-146947, 2021, doi: 10.1109/ACCESS.2021.3124050.
- [5] B. C. Kim, Y. Yu, and Y. M. Ro, "ROBUST DECISION-BASED BLACK-BOX ADVERSARIAL ATTACK VIA COARSE-TO-FINE RANDOM SEARCH," in Proc. 2021 IEEE International Conference on Image Processing (ICIP), 2021, <https://doi.org/10.1109/ICIP42928.2021.9506464>.
- [6] C. Wu et al., "PRADA: Practical Black-box Adversarial Attacks against Neural Ranking Models," ACM Trans. Inf. Syst., vol. 41, no. 4, Art. 89, Apr. 2023, <https://doi.org/10.48550/arXiv.2204.01321>.
- [7] C. Hu, L. Huang, and R. Shi, "Fool a Hashing-Based Video Retrieval System by Perturbing the Last 8 Frames of a Video," in J. Yao et al. (eds.), The International Conference on Image, Vision and Intelligent Systems (ICIVIS 2021), Lecture Notes in Electrical Engineering, vol. 813, pp. xxx-xxx, 2021, doi: 10.1007/978-981-16-6963-7_100.

BIBLIOGRAPHY

- [8] X. Wei et al., "Transferable Adversarial Attacks for Image and Video Object Detection," in Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19), 2019, doi: 10.24963/ijcai.2019/77.
- [9] M. Alzantot et al., "GenAttack: Practical Black-box Attacks with Gradient-Free Optimization," <https://arxiv.org/abs/1805.11090>, Jul. 1, 2019.
- [10] Z. Wei et al., "Boosting the Transferability of Video Adversarial Examples via Temporal Translation," <https://arxiv.org/abs/2110.09075>, Dec. 28, 2021.
- [11] H. Zhang, L. Zhu, Y. Zhu, and Y. Yang, "Motion-Excited Sampler: Video Adversarial Attack with Sparked Prior," ReLER, University of Technology Sydney, NSW, Australia; Amazon Web Services, 2022.
- [12] F. Mumcu, K. Doshi, and Y. Yilmaz, "Adversarial Machine Learning Attacks Against Video Anomaly Detection Systems," in Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2022, <https://doi.org/10.48550/arXiv.2204.03141>
- [13] Z. Chen, L. Xie, S. Pang, Y. He and Q. Tian, "Appending Adversarial Frames for Universal Video Attack," 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2021, pp. 3198-3207, doi: 10.1109/WACV48630.2021.00324.
- [14] K. Chen et al., "Attacking Video Recognition Models with Bullet-Screen Comments," arXiv:2110.15629v1 [cs.CV], Oct. 29, 2021. Available: <https://arxiv.org/abs/2110.15629v1>
- [15] W. Sultani, C. Chen and M. Shah, "Real-World Anomaly Detection in Surveillance Videos," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 6479-6488, doi: 10.1109/CVPR.2018.00678.
- [16] S. Li, A. Aich, S. Zhu, M. S. Asif, C. Song, A. K. Roy-Chowdhury, and S. Krishnamurthy, "Adversarial Attacks on Black Box Video Classifiers: Leveraging the Power of Geometric Transformations," presented at the 35th Conference on Neural

BIBLIOGRAPHY

Information Processing Systems (NeurIPS 2021), Sydney, Australia, Oct. 5, 2021, <https://arxiv.org/abs/2110.01823>.

[17] Yin, M.; Xu, Y.; Hu, T.; Liu, X. A Robust Adversarial Example Attack Based on Video Augmentation. *Appl. Sci.* 2023, 13, 1914. <https://doi.org/10.3390/app13031914>.

[18] L. Jiang, X. Ma, S. Chen, J. Bailey, and Y.-G. Jiang, "Black-box Adversarial Attacks on Video Recognition Models," <https://arxiv.org/abs/1904.05181>, Jun. 28, 2019.

[19] A. Guesmi et al., "Defensive Approximation: Securing CNNs using Approximate Computing," in Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '21), Virtual, USA, Apr. 19–23, 2021, pp. 14, <https://dl.acm.org/doi/10.1145/3445814.3446747>.

[20] J. Renkhoff, W. Tan, A. Velasquez, W. Y. Wang, Y. Liu, J. Wang, S. Niu, L. B. Fazlic, G. Dartmann, H. Song, "Exploring Adversarial Attacks on Neural Networks: An Explainable Approach," <https://arxiv.org/abs/2303.06032>, Mar. 8, 2023.

[21] A. Jan and G. M. Khan, "Real-world malicious event recognition in CCTV recording using Quasi-3D network," *J. Ambient Intell. Humaniz. Comput.*, vol. Online First, Jan. 2022. <https://link.springer.com/article/10.1007/s12652-022-03702-6>.

[22] A. Sharma, Y. Bian, P. Munz, and A. Narayan, "Adversarial Patch Attacks and Defences in Vision-Based Tasks: A Survey," <https://arxiv.org/pdf/2206.08304.pdf>, Jun. 16, 2022.

[23] X. Wei, J. Zhu, and H. Su, "Sparse Adversarial Perturbations for Videos," <https://arxiv.org/pdf/1803.02536.pdf>, Mar. 7, 2018.

[24] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," <https://arxiv.org/pdf/1608.04644.pdf>, Mar. 22, 2017.

- [25] A. M. A. Al-salam, S. Ahmed, and F. Fadhil, "A Study of the Effects of Gaussian Noise on Image Features," *Kirkuk University Journal-Scientific Studies*, https://kujss.uokirkuk.edu.iq/article_124648.html, Apr. 2016.
- [26] J. Yan, X. Deng, H. Yin, and W. Ge, "On Procedural Adversarial Noise Attack And Defense," Department of Information and Communication Engineering, Tongji University, Shanghai, China, <https://arxiv.org/pdf/2108.04409v2.pdf>, Aug. 27, 2021.
- [27] Patel, Pooja & Bhandari, Arpana. (2019). A Review on Image Contrast Enhancement Techniques. *SMART MOVES JOURNAL IJOSCIENCE*. 5. 5. 10.24113/<http://dx.doi.org/10.24113/ijoscience.v5i7.217>.
- [28] B. Yang, K. Xu, H. Wang, and H. Zhang, "Random Transformation of Image Brightness for Adversarial Attack," <https://arxiv.org/abs/2101.04321>, Jan. 12, 2021.
- [29] Z. A. A. Al Qadi, "Salt and Pepper Noise: Effects and Removal," *International Journal on Electrical Engineering and Informatics*, Jul. 2018.
- [30] N. P. Bhosale, R. Manza, and K. Kale, "Analysis of Effect of Gaussian, Salt and Pepper Noise Removal from Noisy Remote Sensing Images," August 2014.
- [31] G. M. Mahesh, "Image Deblurring Techniques - A Detail Review," January 2018.
- [32] N. Carlini and D. Wagner, "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods," <https://arxiv.org/pdf/1705.07263.pdf>, Nov. 1, 2017.
- [33] A. Labrada, B. Bustos, and I. Sipiran, "A Convolutional Architecture for 3D Model Embedding," <https://arxiv.org/pdf/2103.03764.pdf>, Mar. 5, 2021.
- [34] A. Rayhan, D. Gross, and S. Rayhan, "Exploring Advancements in AI Algorithms, Deep Learning, Neural Networks, and Their Applications in Various Fields," Preprint, August 2023 <http://dx.doi.org/10.13140/RG.2.2.18923.31522>.
- [35] S. O. C. Morales and S. J. Cox, "On the estimation and the use of confusion-matrices for improving ASR accuracy," in *Interspeech*, 2009. [Online]. Available: <https://api.semanticscholar.org/CorpusID:6316280>.