
HELLO FRIEND.

Watching the Watchers:

Practical Video Identification Attack in LTE Networks

Ryan Abdi

Tarbiat Modares University

In collaboration with Prof. Sharafat

Tarbiat Modares University, May 27, 2023

Introduction

Privacy Threat: Identifying User's Video Viewing History

- **Increasing Usage:** Cellular Networks for Mobile Video Streaming
- **Privacy Concerns:**
  - Video Viewing History: Unveiling Private Information: Political, Financial, and Personal Interests
- **Privacy Threat:**
  - Attacker's Ability: Identifying Videos Watched by Individuals
  - Serious Implications: Privacy Threat to User's Privacy

Introduction

Video Identification Attack: Exploiting Encrypted Traffic

- **Video Identification Attack:** Adversary Identifying Videos through Encrypted Traffic Analysis
- **Exploiting Encrypted Video Streams:**
  - Identifiable Fingerprint: Operating Logic of HTTP Adaptive Streaming (HAS)
- **Segmenting Video Content:**
  - Smaller Chunks Varying in Size
  - Formation of Unique Patterns: Series of Segmented Chunks

Introduction

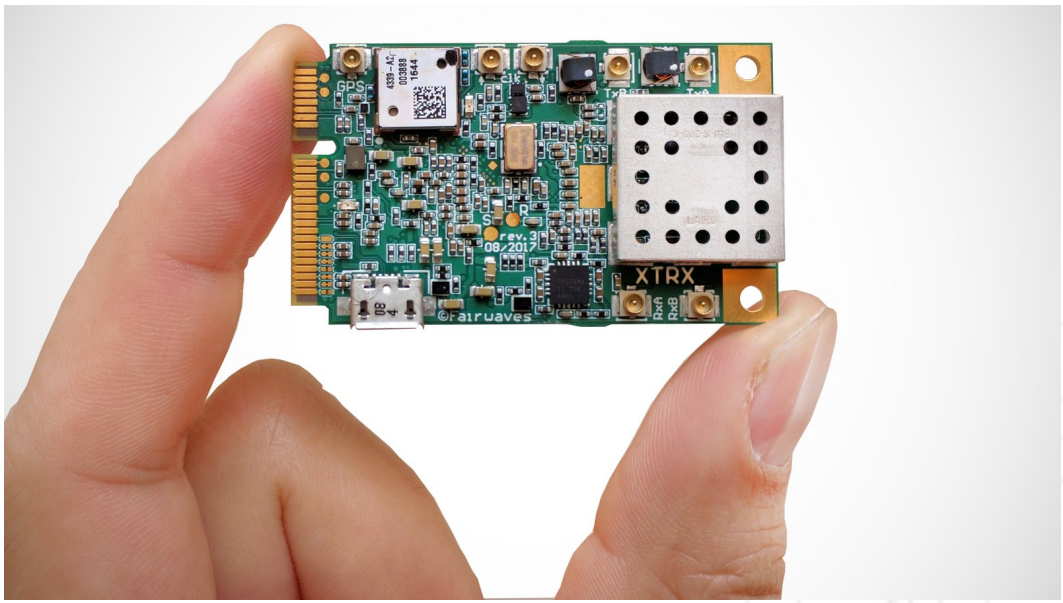
Video Identification Attack On Wired Networks

- **History of Video Identification Attacks:** Primarily Targeting Wired Networks
- **Assumptions:** Adversary's Strong Requirements for Attack:
  - Direct network infrastructure access
  - Execution of malicious apps or websites on victim's device

Introduction

Video Identification Attacks on Wireless Networks

- **Wireless Networks:** Broadcasting Radio Signals Over the Air
- **Software-Defined Radio (SDR):** Enabling Adversarial Monitoring Abilities



Introduction

Unprivileged Video Identification Attacks

Adversary Power in Wireless Networks

If this adversary is able to extract **effective features** from these signals that contribute to **identifying streamed videos**, they become a strong adversary who can carry out a video identification attack
without any privilege

Introduction

Video Identification Attack in LTE Networks

- **Purpose:** Presenting the First Study of Video Identification Attack in LTE Networks
- **Investigating Feasibility:** Can an adversary:
  1. Pinpoint users watching specific videos?
  2. Identify the exact titles?
- **Assumptions:**
  - Unprivileged Adversary: No access to victims' devices or cell towers
  - Equipment: Utilizing SDR
- **Implications:**
  - Tracking Illicit Content: Potential for identifying users watching illegal or sensitive videos

Introduction

Technical Challenges in Video Identification Attacks in LTE Networks

- **Challenge 1: Limited Monitoring Capability**
  - Difficulty in Determining Traffic Type:
    Identifying video sources (e.g., YouTube, Netflix) and constructing accurate video fingerprints
- **Challenge 2: Dynamic Traffic Collection**
  - Complexity in Collecting Victim's Traffic:
    Changing users identifiers during video playback caused by the interworking of HAS and LTE network operations
- **Challenge 3: Carrier Aggregation:**
  - Aggregating Video Traffic:
    Dealing with carrier aggregation (CA) logic employed by LTE network operators for efficient bandwidth utilization

Introduction

Proposed Solutions to Address Video Identification Challenges

- **Solution 1: Distinctive Characteristics Identification**
  - Identifying Video Service Provider:
    Utilizing Decision Tree Classifiers
  - Identifying Video Title:
    Utilizing Convolutional Neural Network (CNN)
- **Solution 2: Linking Changing Identifiers**
  - Exploiting Unencrypted Information Transmitted to Victims
- **Solution 3: Channel Monitoring Optimization**
  - Heuristic Method for Estimating Video Traffic Volume
  - Monitoring a Single Channel Instead of All Channels for CA

Background

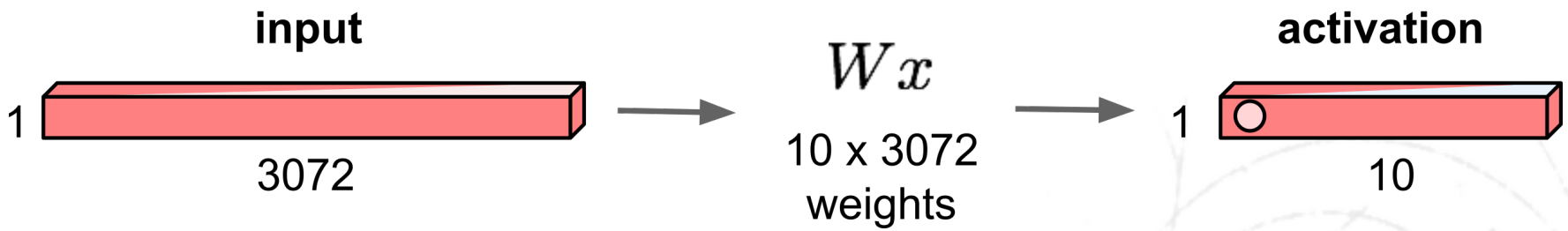
Convolutional Neural Network

- **Architecture**
  - Convolution Layer
  - Pooling Layer
  - Fully Connected Layer
- **Setting Up The Data and The Model**
  - Dropout
- **Training**
  - Parameters
  - Loss Function

Background

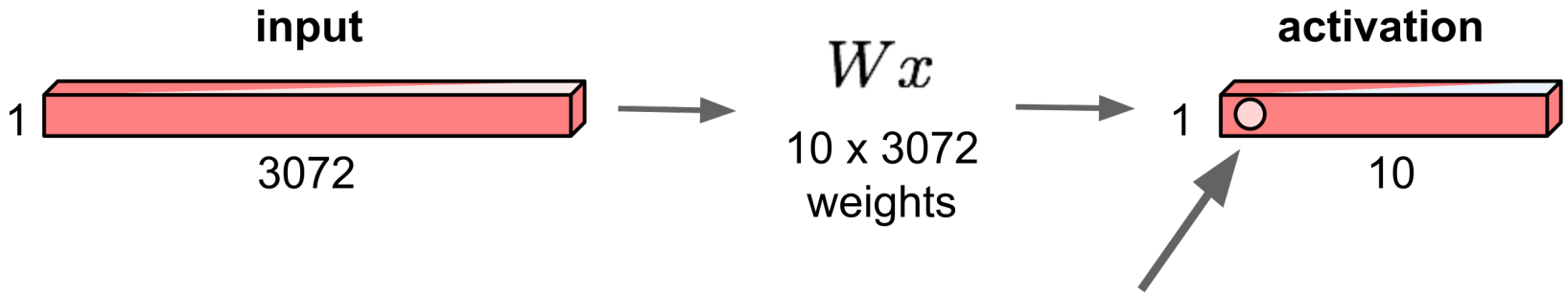
CNN- Architecture: Fully Connected Layer (I)

32x32x3 image → stretch to 3072 x 1



Background

CNN- Architecture: Fully Connected Layer (II)

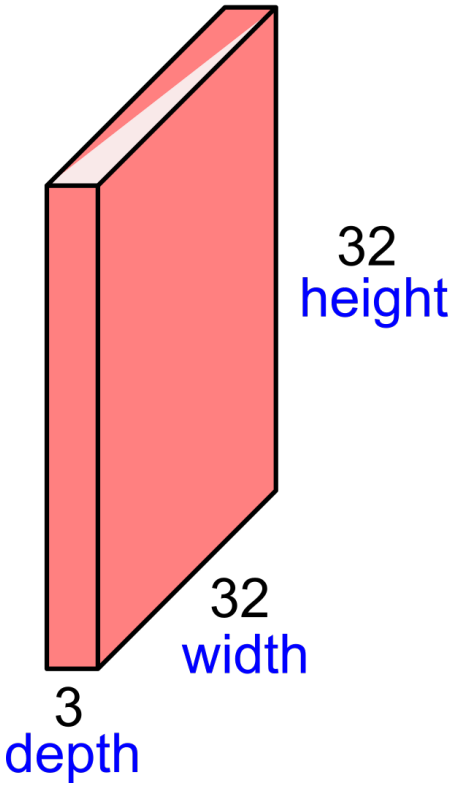


1 number:
the result of taking a dot product
between a row of W and the input
(a 3072-dimensional dot product)

Background

CNN- Architecture: Convolution Layer (I)

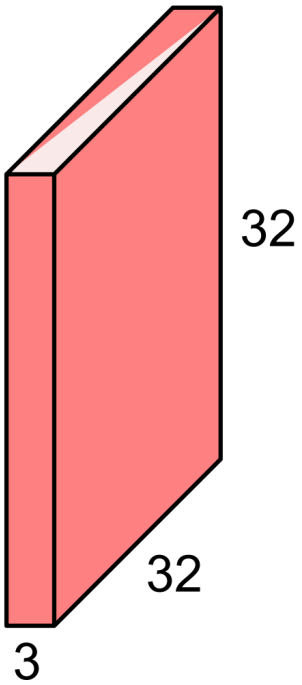
32x32x3 image → preserve spatial structure



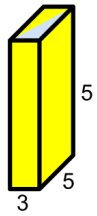
Background

CNN- Architecture: Convolution Layer (II)

32x32x3 image



5x5x3 filter



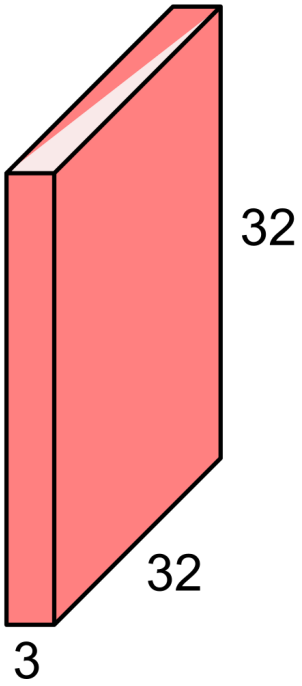
Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”

Background

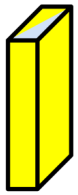
CNN- Architecture: Convolution Layer (III)

Filters always extend the full depth of the input volume

32x32x3 image



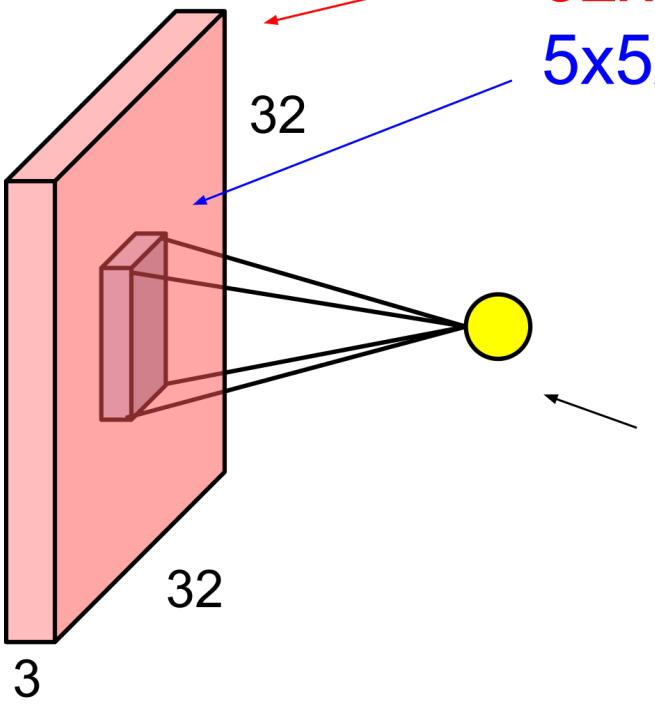
5x5x3 filter



Convolve the filter with the image i.e. “slide over the image spatially, computing dot products”

Background

CNN- Architecture: Convolution Layer (IV)

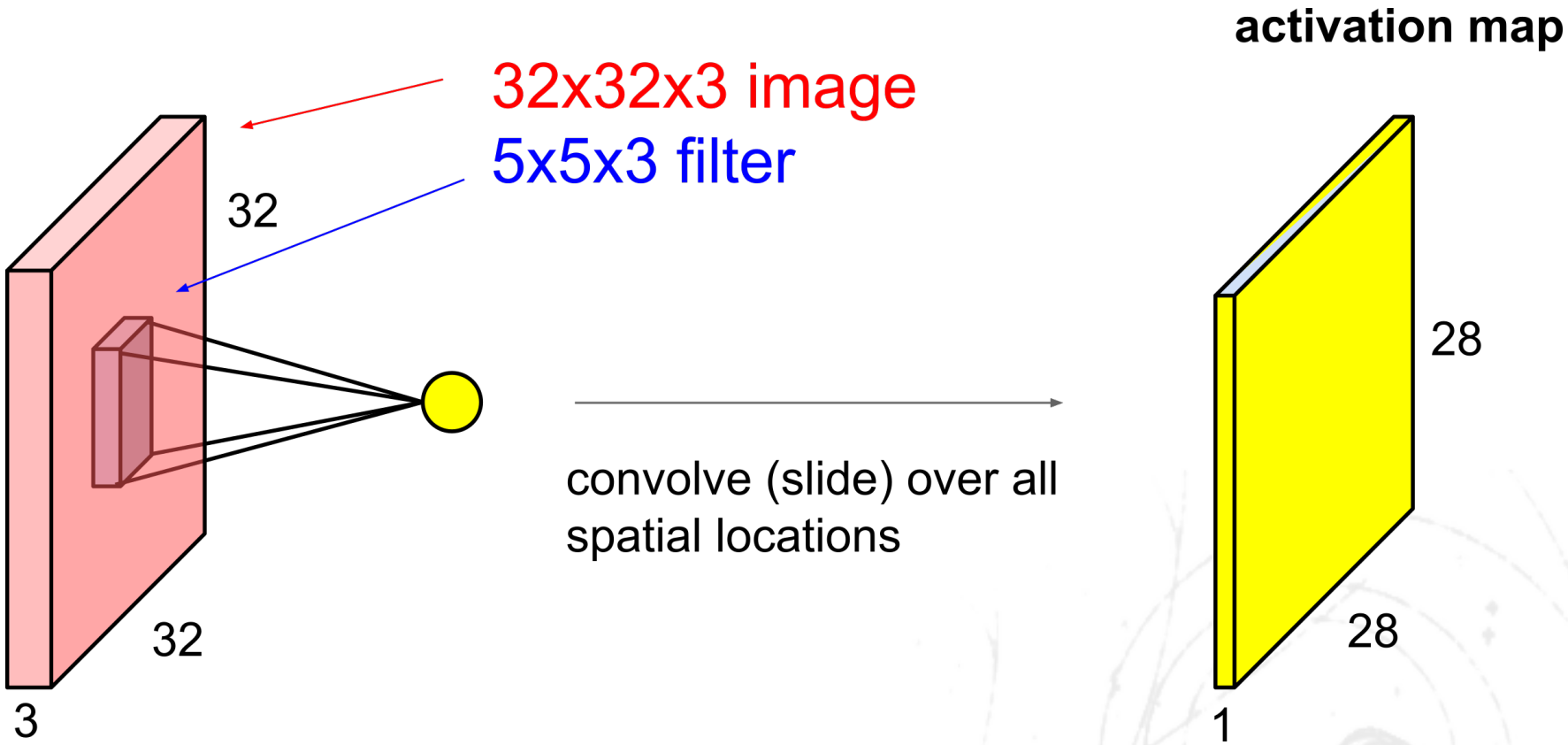


1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image (i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

Background

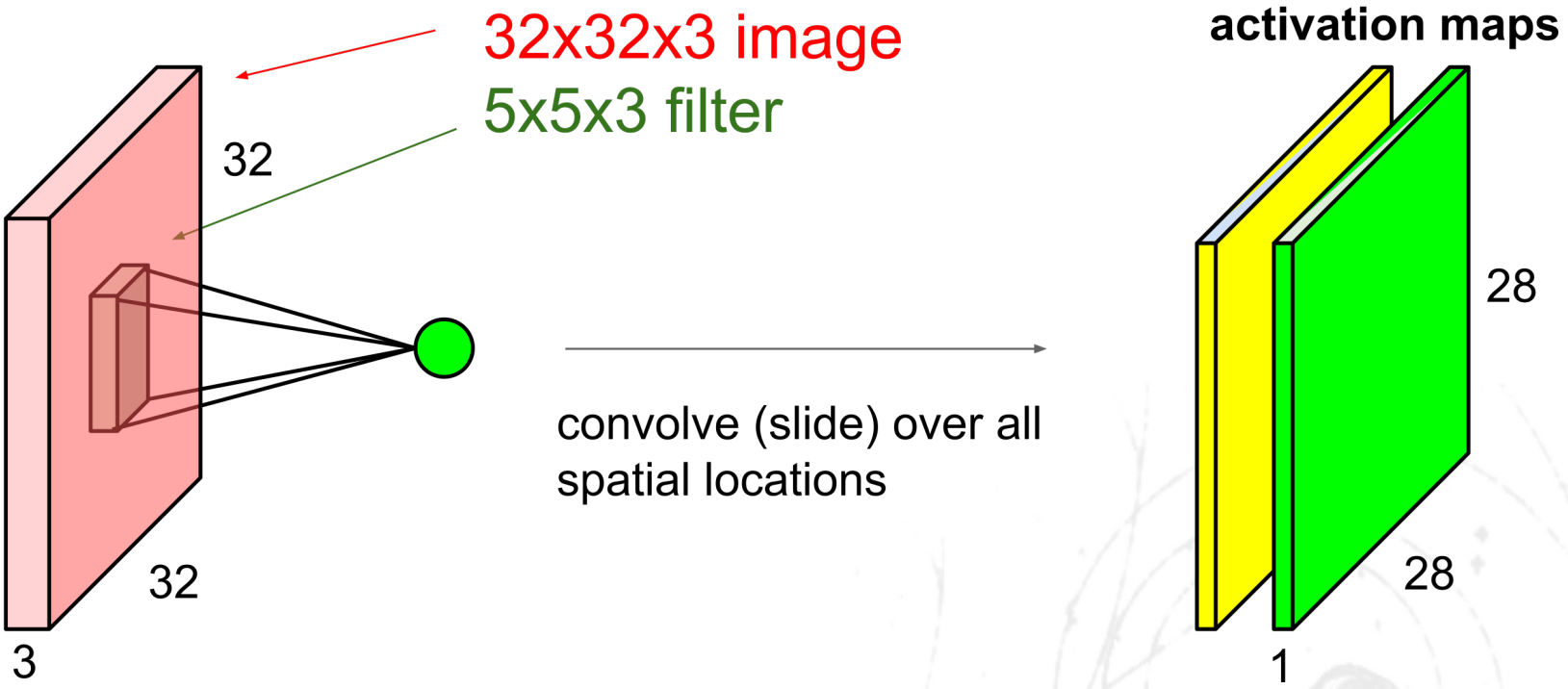
CNN- Architecture: Convolution Layer (V)



Background

CNN- Architecture: Convolution Layer (VI)

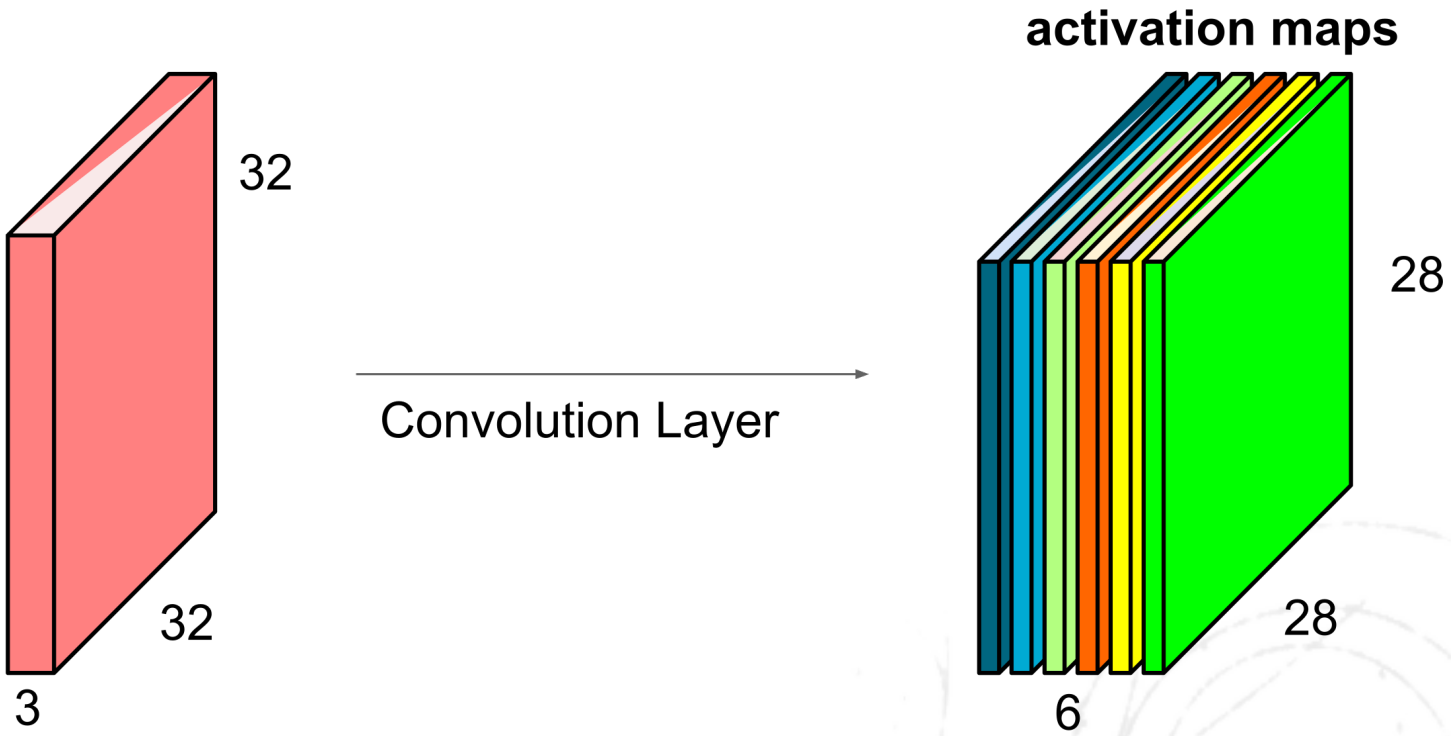
consider a second, green filter



Background

CNN- Architecture: Convolution Layer (VII)

For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:

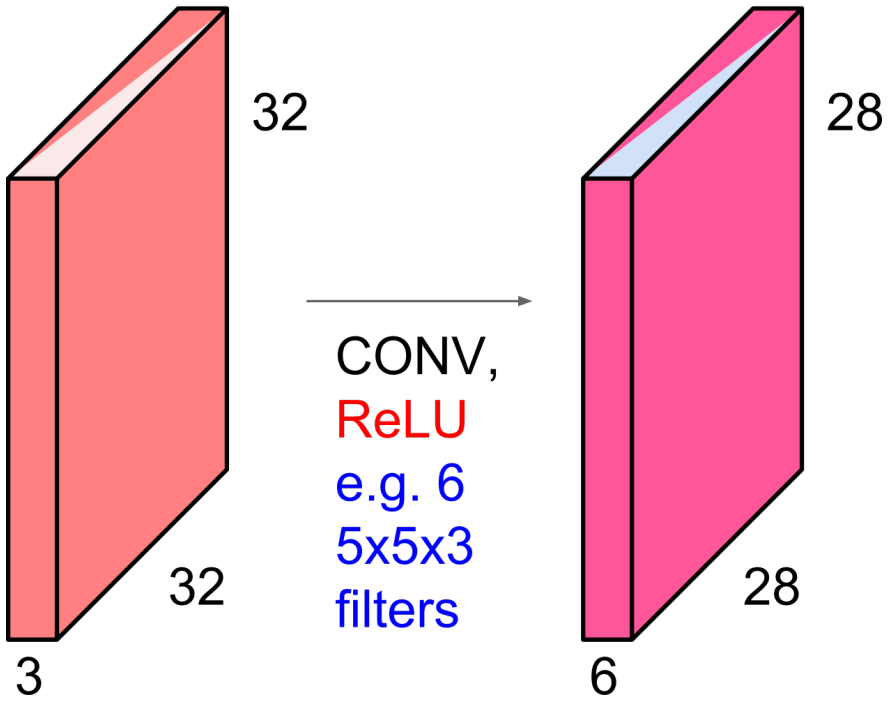


We stack these up to get a "new image" of size 28x28x6!

Background

CNN- Architecture: Convolution Layer (VIII)

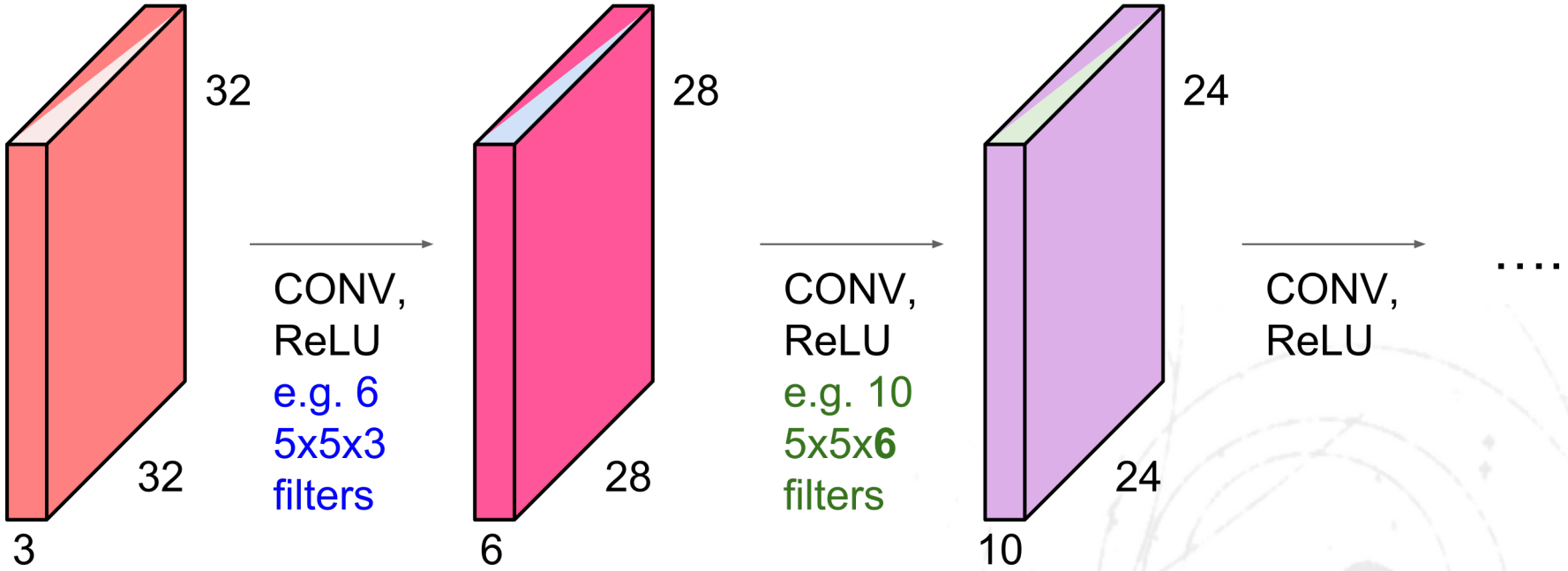
Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



Background

CNN- Architecture: Convolution Layer (VIIII)

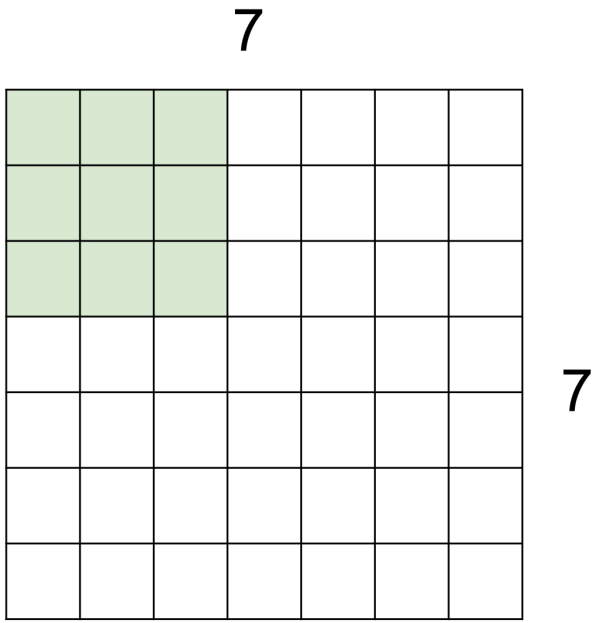
Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



Background

CNN- Architecture: Dimensions (I)

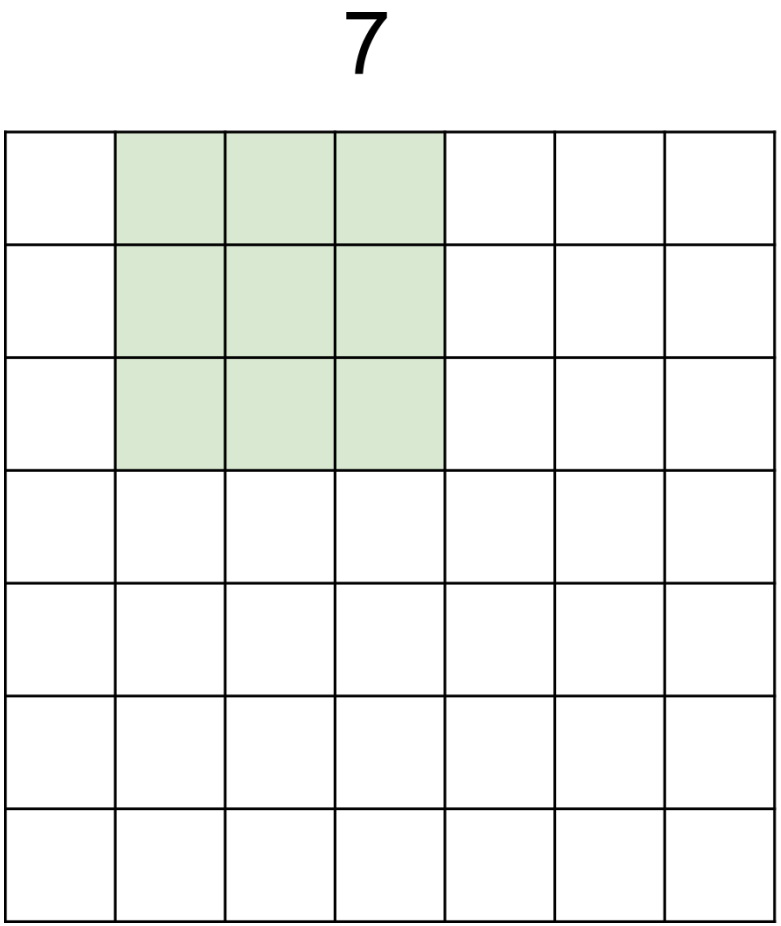
A closer look at spatial dimensions:



7x7 input (spatially)
assume 3x3 filter

Background

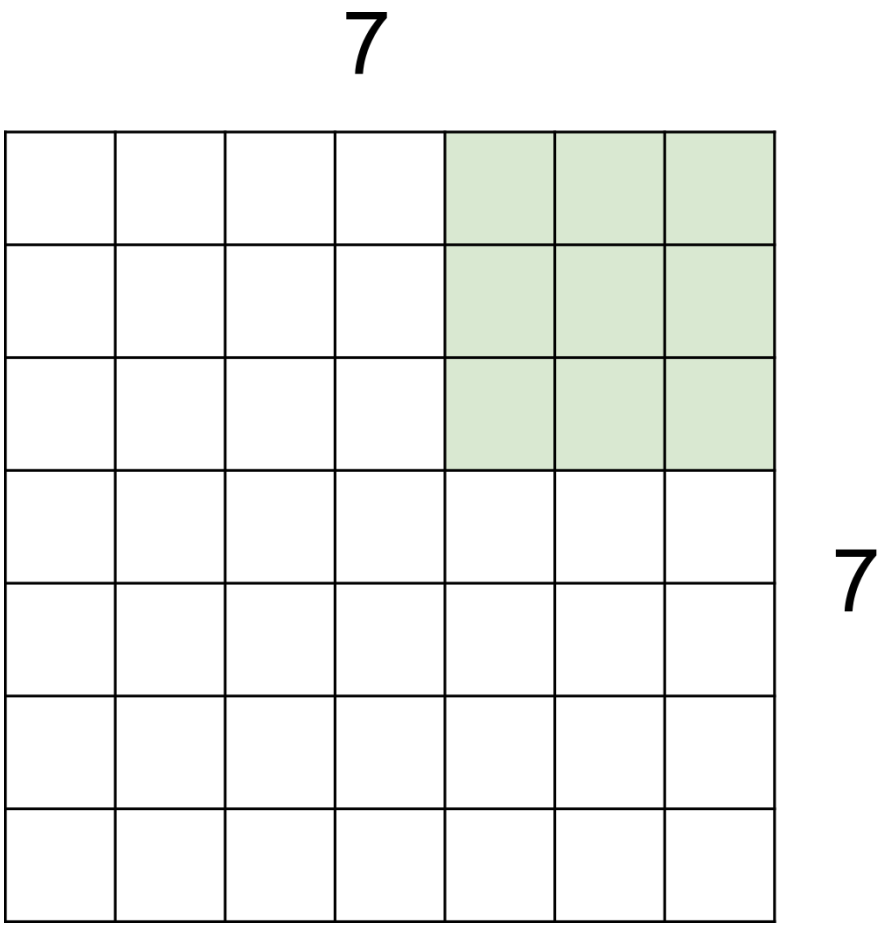
CNN- Dimensions (II)



7x7 input (spatially)
assume 3x3 filter

Background

CNN- Architecture: Dimensions (III)

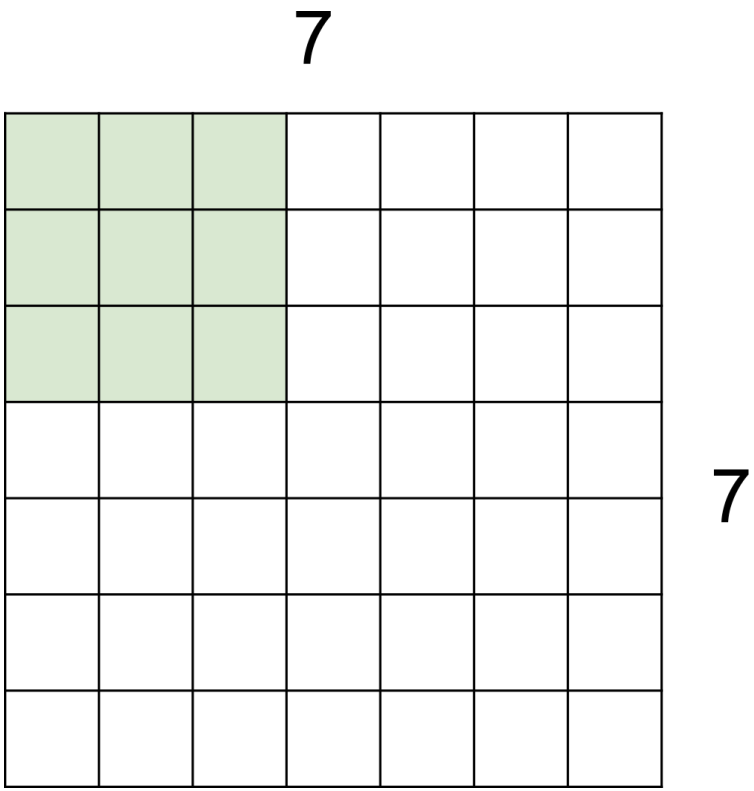


7x7 input (spatially)
assume 3x3 filter

=> 5x5 output

Background

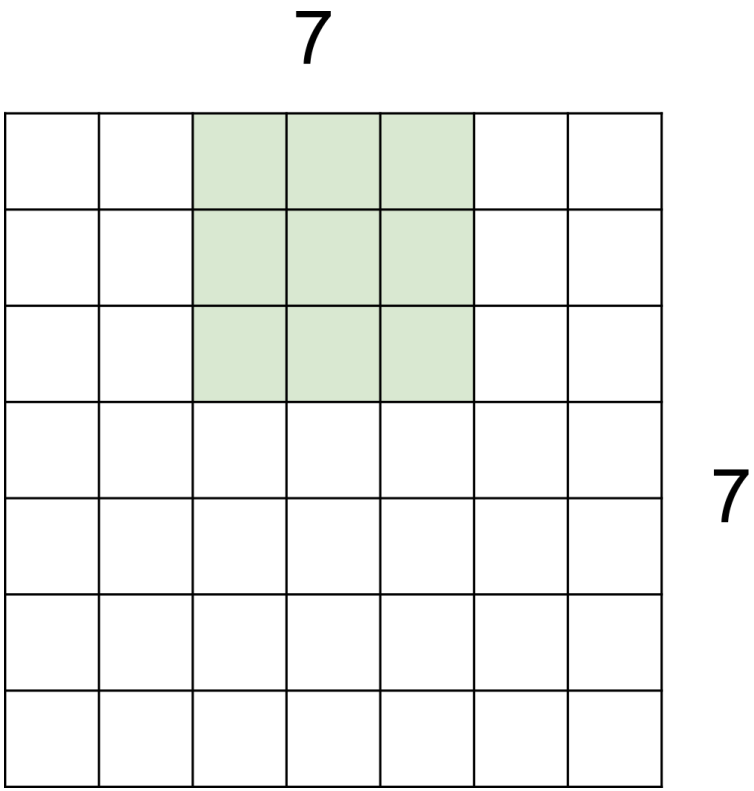
CNN- Strides (I)



7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**

Background

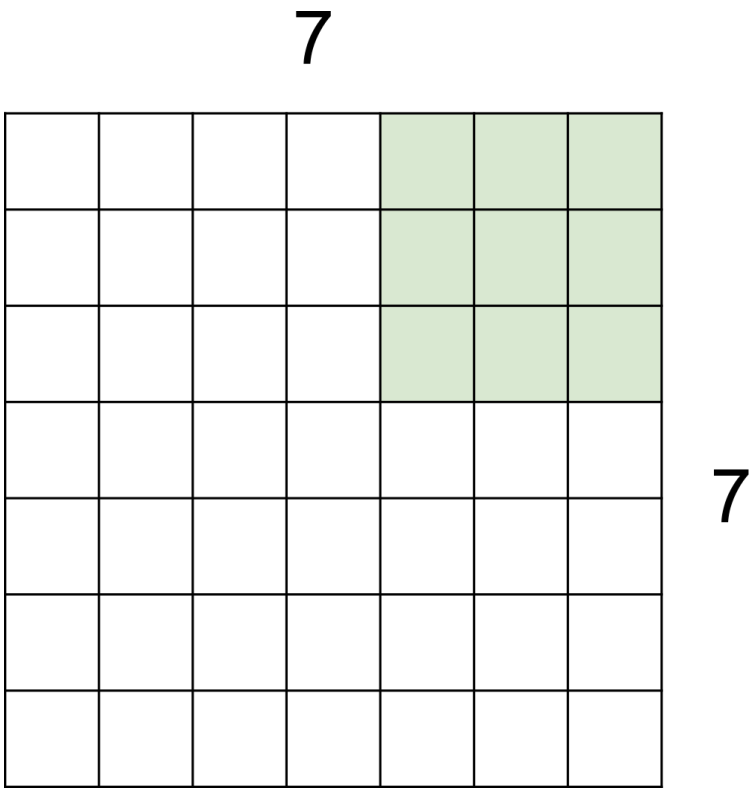
CNN- Architecture: Strides (II)



7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**

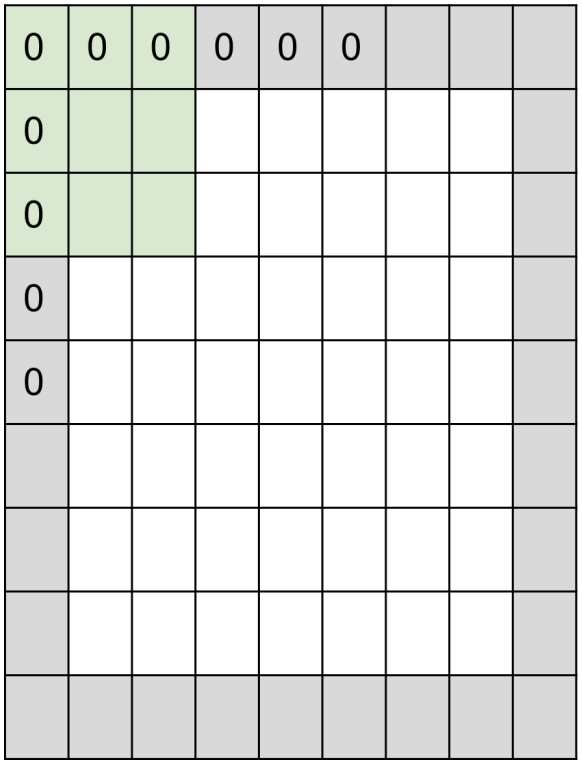
Background

CNN- Architecture: Strides (III)



7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**
=> 3x3 output!

In practice: Common to zero pad the border



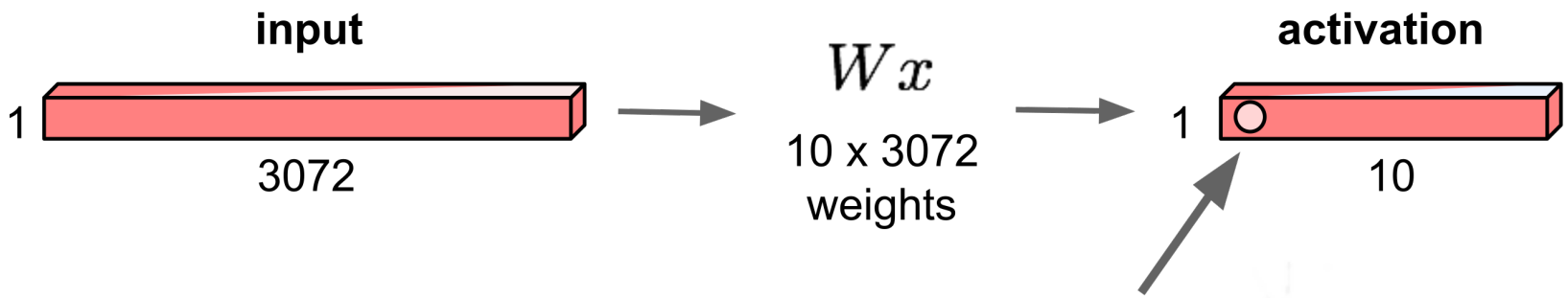
e.g. input 7x7
3x3 filter, applied with **stride 1**
pad with 1 pixel border => what is the output?

7x7 output!

Background

CNN- Architecture: Intuition (I)

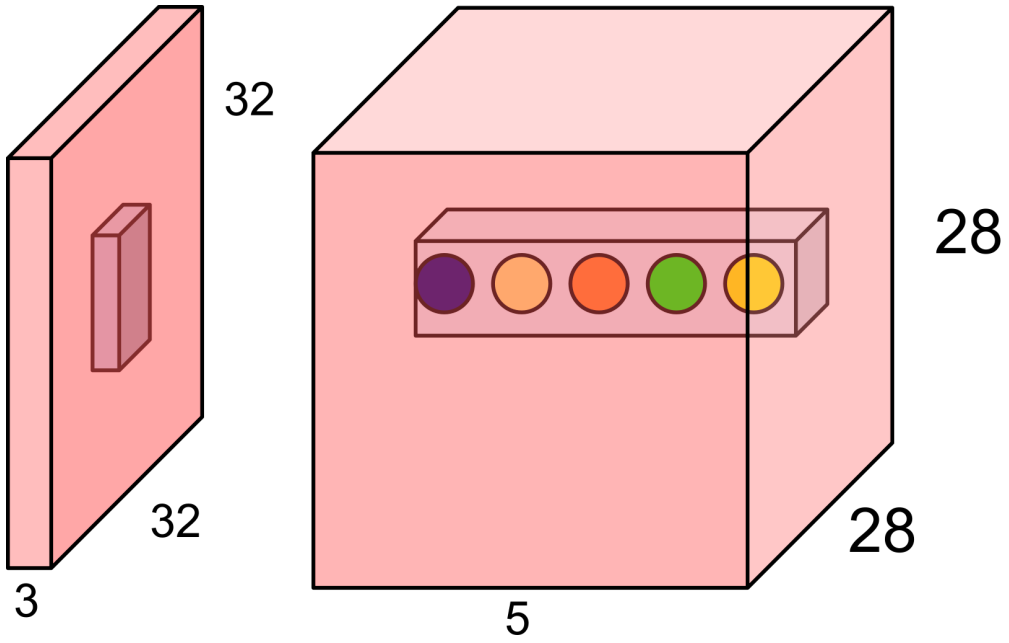
Each neuron looks at the full input volume



1 number:
the result of taking a dot product between a row of W and the input (a 3072-dimensional dot product)

Background

CNN- Architecture: Intuition (II)

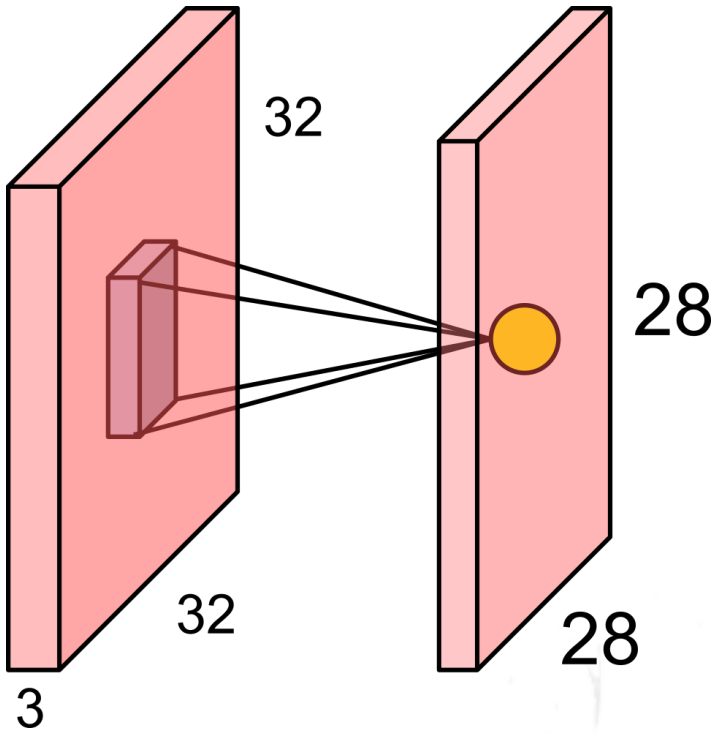


E.g. with 5 filters,
CONV layer consists of
neurons arranged in a 3D grid
(28x28x5)

There will be 5 different
neurons all looking at the same
region in the input volume

Background

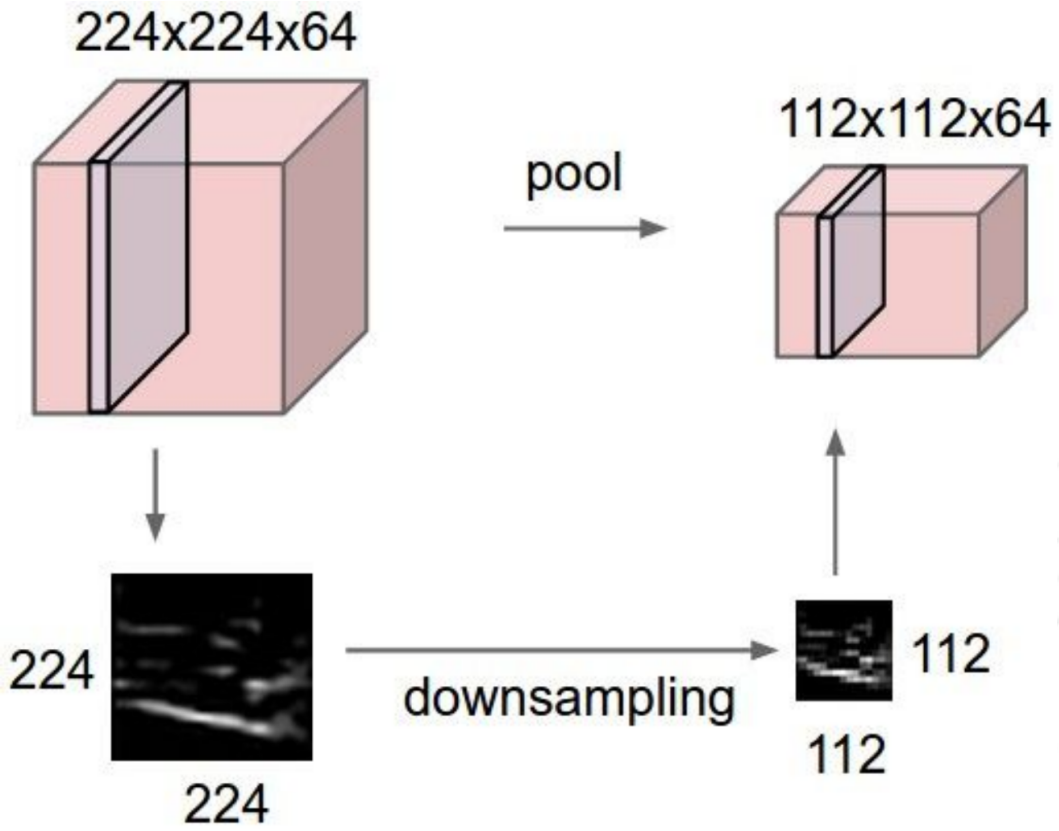
CNN- Architecture: Intuition (III)



Background

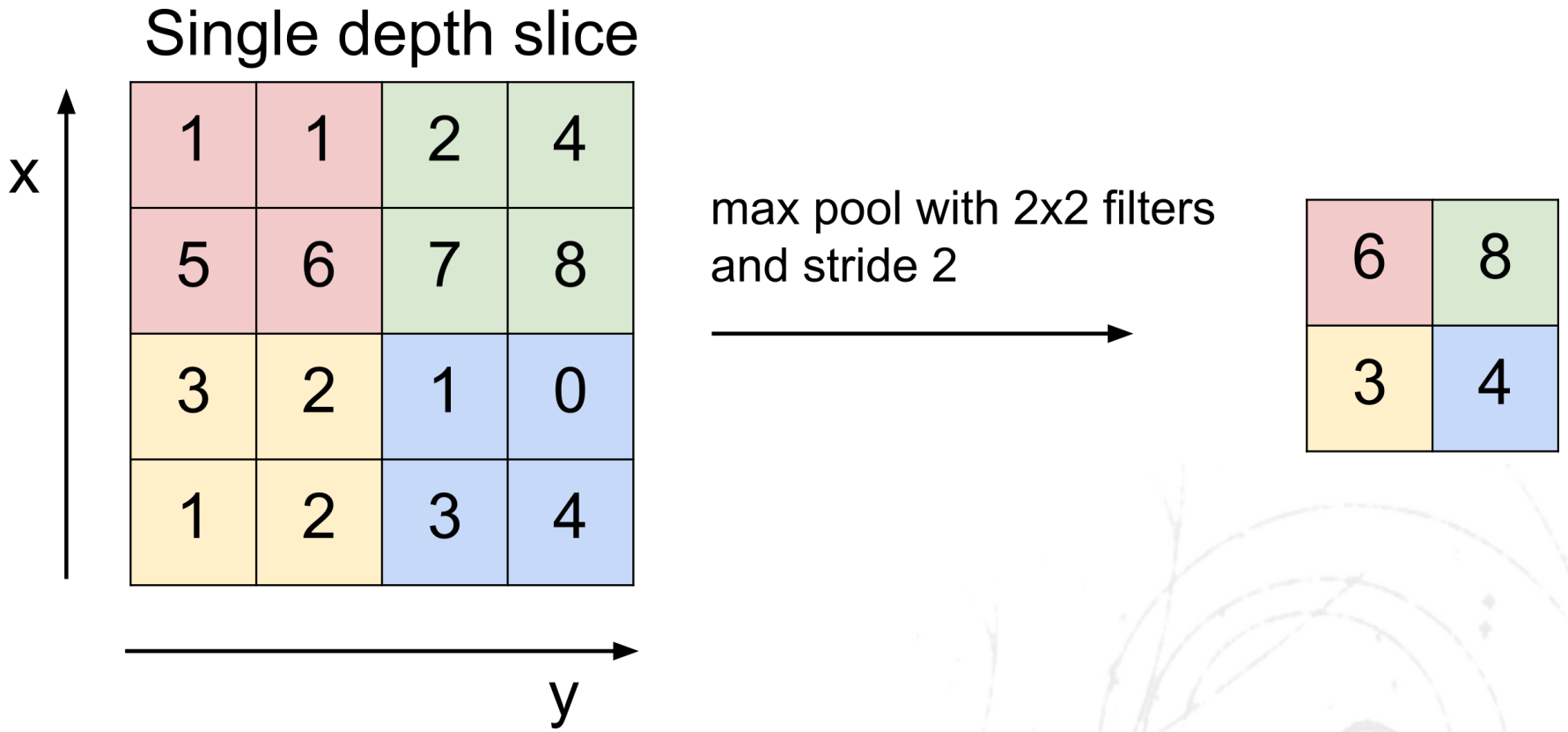
CNN- Architecture: Pooling Layer

- Smaller and More Manageable Representations
- Activation Map Independence



Background

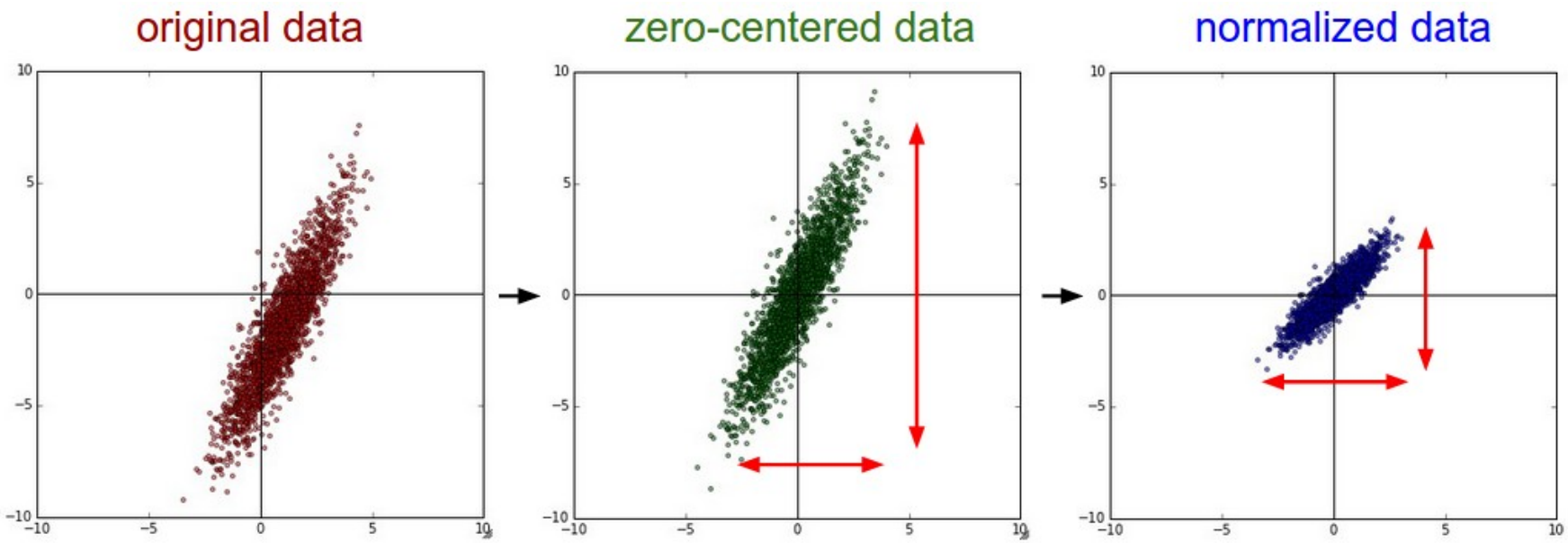
CNN- Architecture: Max Pooling



Background

CNN- Setting Up The Data and The Model: Data

- **Data Preprocessing**
  - Mean subtraction
  - Normalization
  - PCA and Whitening



Background

CNN- Setting Up The Data and The Model: Model (I)

- **Weight Initialization**
  - All Zero Initialization
  - Small Random Numbers
- **Batch Normalization**
  - Robust To Bad Initialization
  - Preprocessing At Every Layer, But Integrated Into the Network Itself!

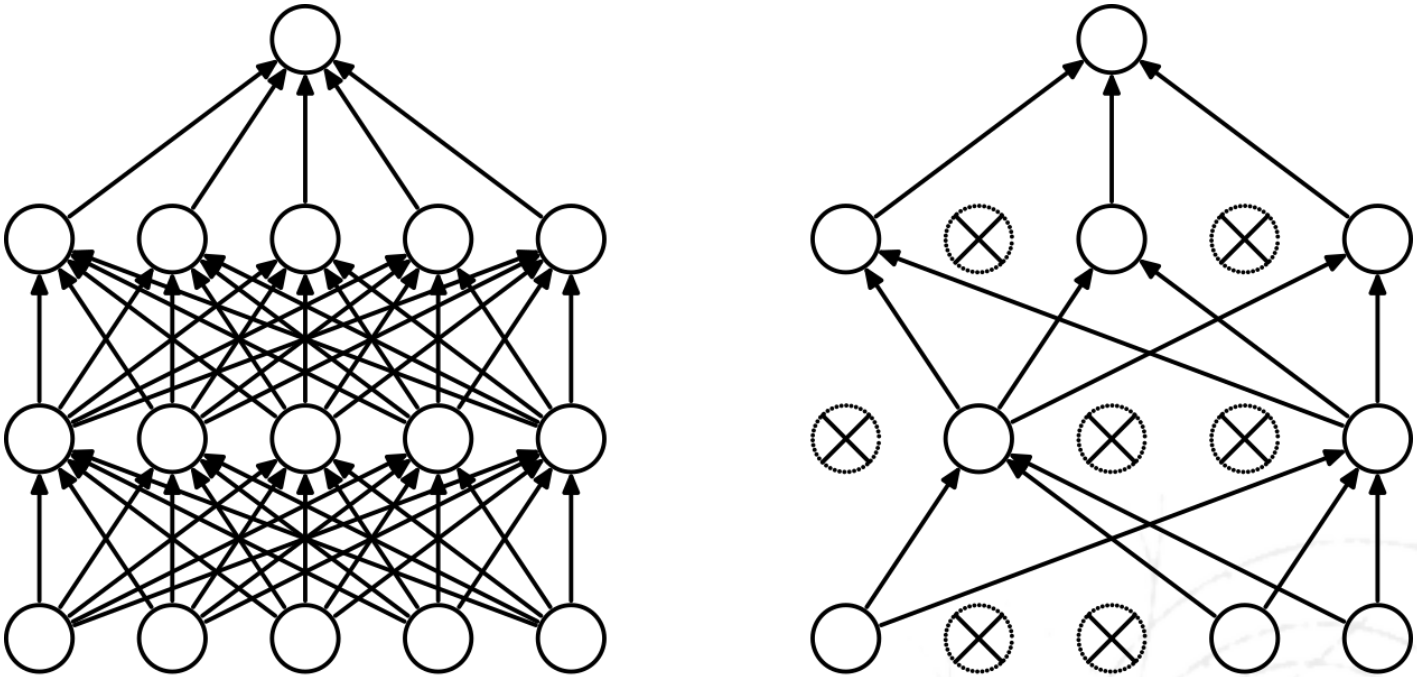
Background

CNN- Setting Up The Data and The Model: Model (II)

- **Regularization**
  Helps Prevent Overfitting
  - Too Much Learning from Training Data
  - Fails To Generalize
  - Dropout
    - Dropping Out Units

Background

CNN- Setting Up The Data and The Model: Dropout (I)



(a) Standard Neural Net

(b) After applying dropout.

CNN- Setting Up The Data and The Model: Dropout (II)

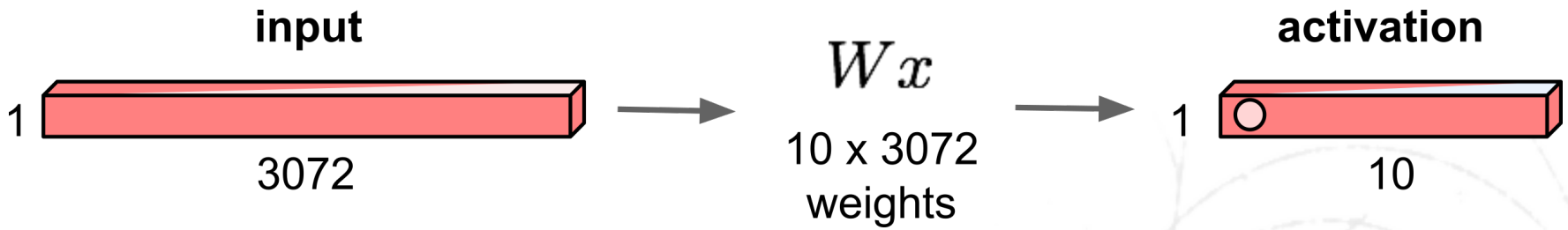
How Dropout Helps Overfitting?

- **Reducing Co-Adaptation**
- **Robust Feature Learning**
- **Ensemble Learning**

Background

CNN- Training: Parameters (I)

- **Supervised**
  - Using Fully Connected Layer at The End
- **Unsupervised**
  - Extracting Features (CNN Minus FCL) + K-Means(Clustering)



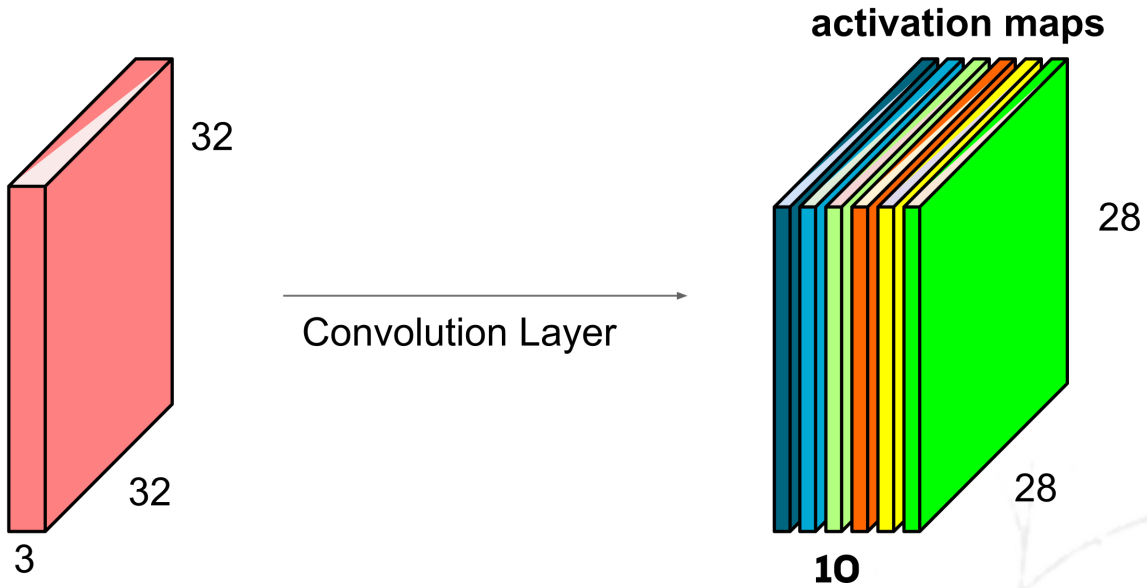
- $\#W = 32 \times 32 \times 3 \times 10 = 30720$

Background

CNN- Training: Parameters (II)

- 10 Filters with Stride 1 and Padding 2

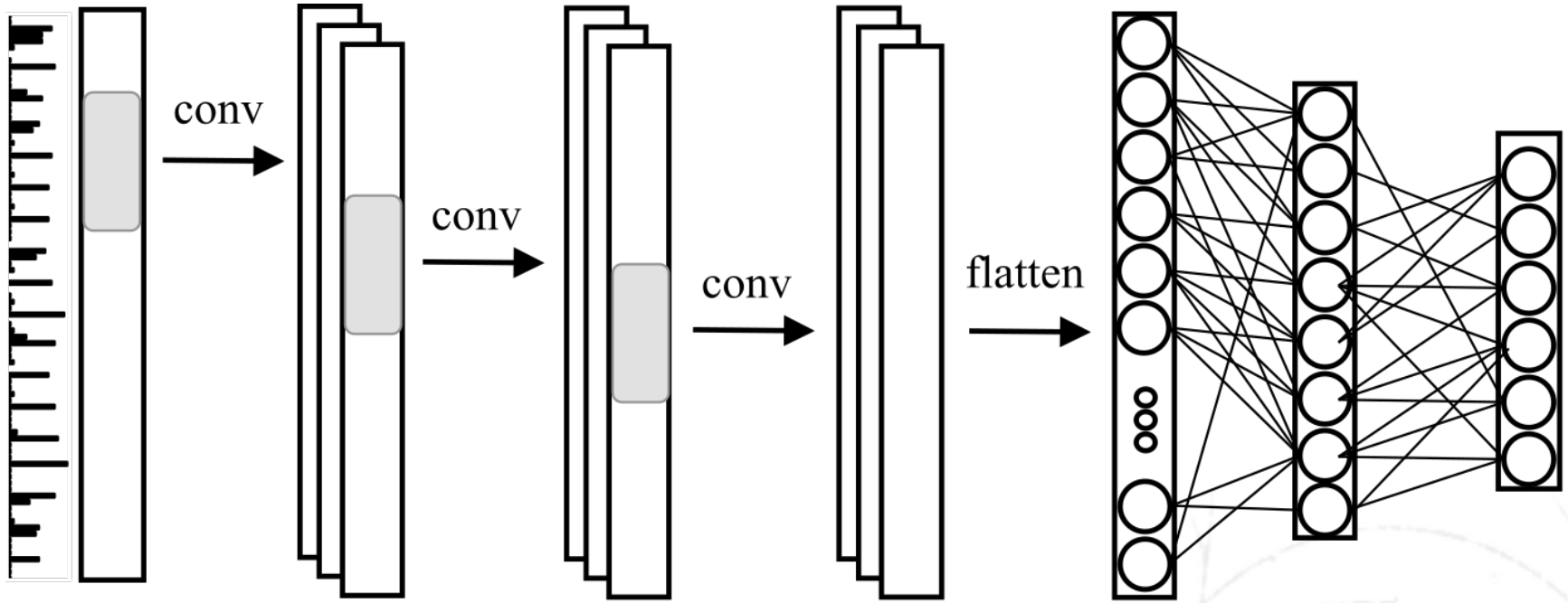
For example, if we had **10** 5x5 filters, we'll get 6 separate activation maps:



- **Each Filter:**
  $\#W_F = 5 \times 5 \times 3 + 1 = 76$
  - +1 Is for Bias
- $\#W = 76 \times 10 = 760$

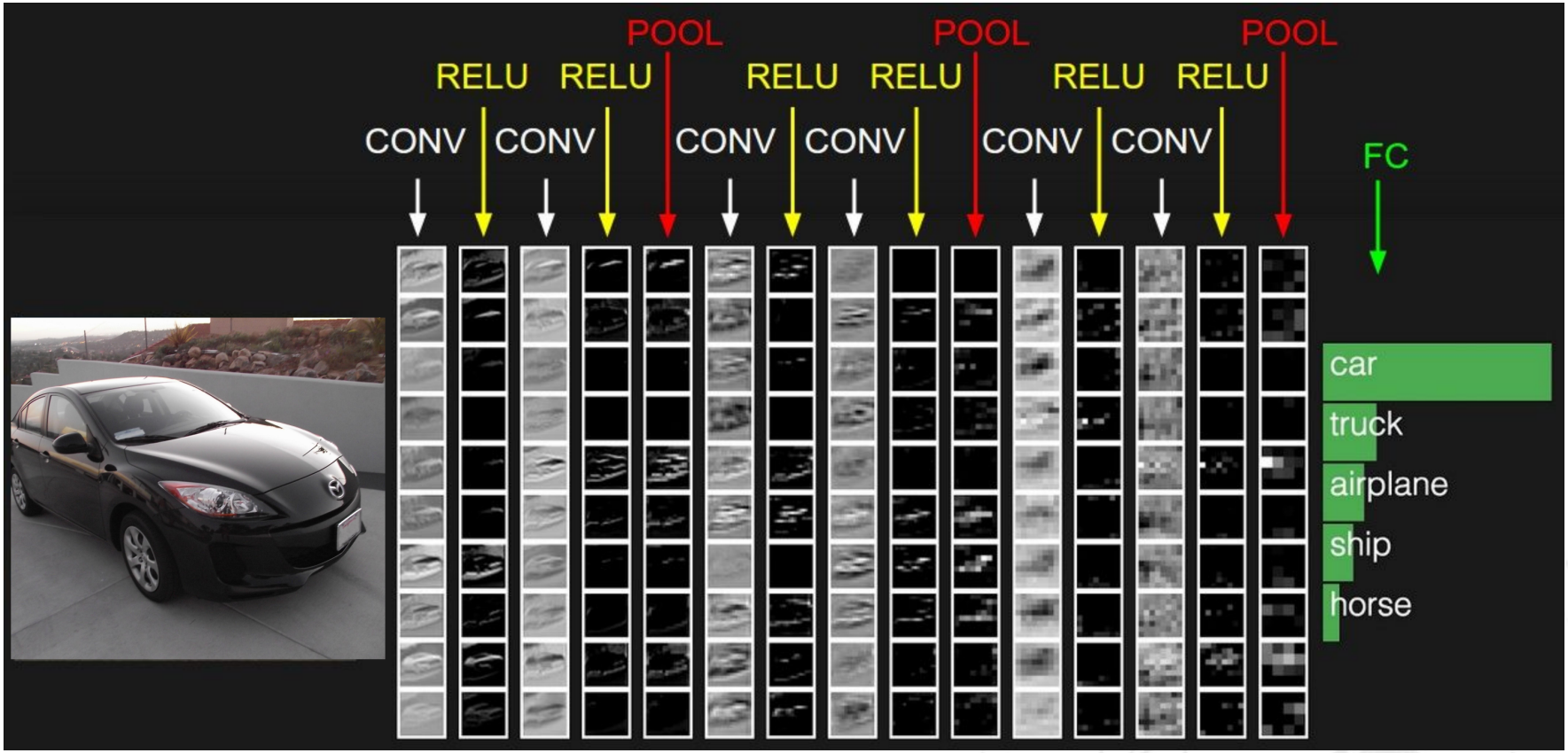
Background

CNN- Example (I)



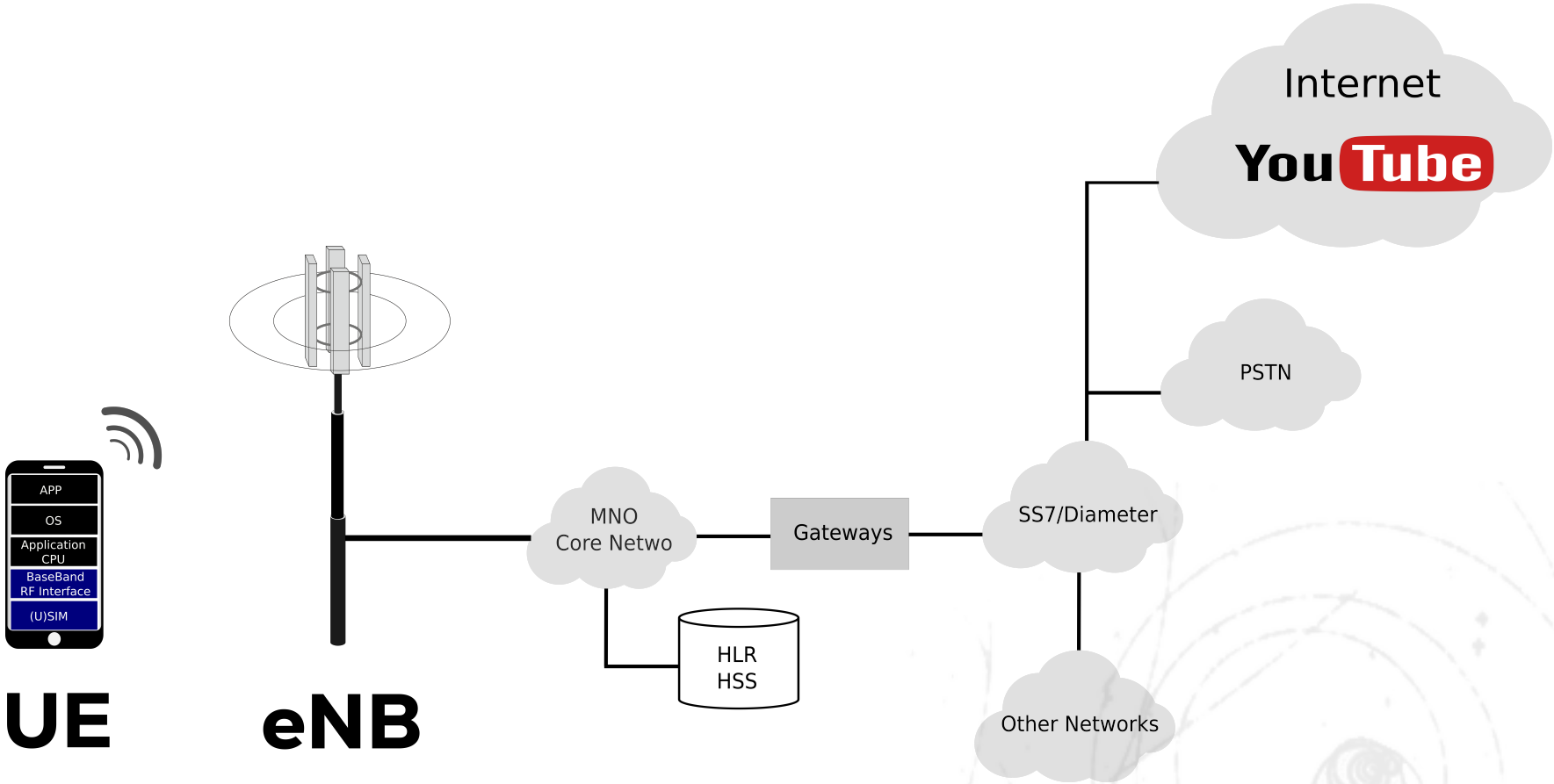
Background

CNN- Example (II)



Background

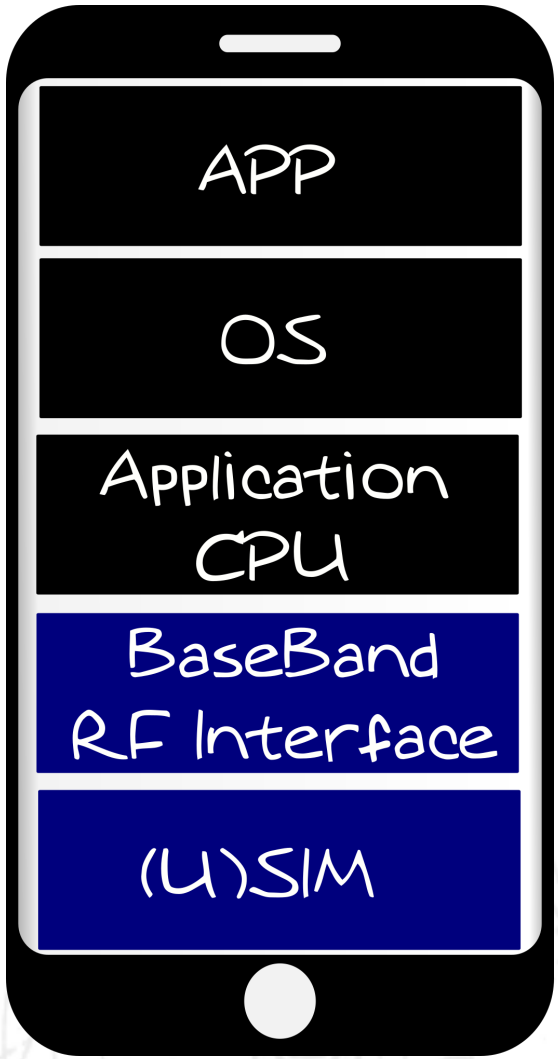
LTE Network Architecture



Background

UE

- (UE)^{4G} communicates with the network and consumes its services.
- The baseband processor implements the mobile protocol stacks
- (SIM)^{2G}, (USIM) $^{3G/4G}$
  - identifies a customer
  - stores the authentication information
    - IMSI
    - secret long-term symmetric key used for encryption and authentication
  - People know your MSISDN



Background

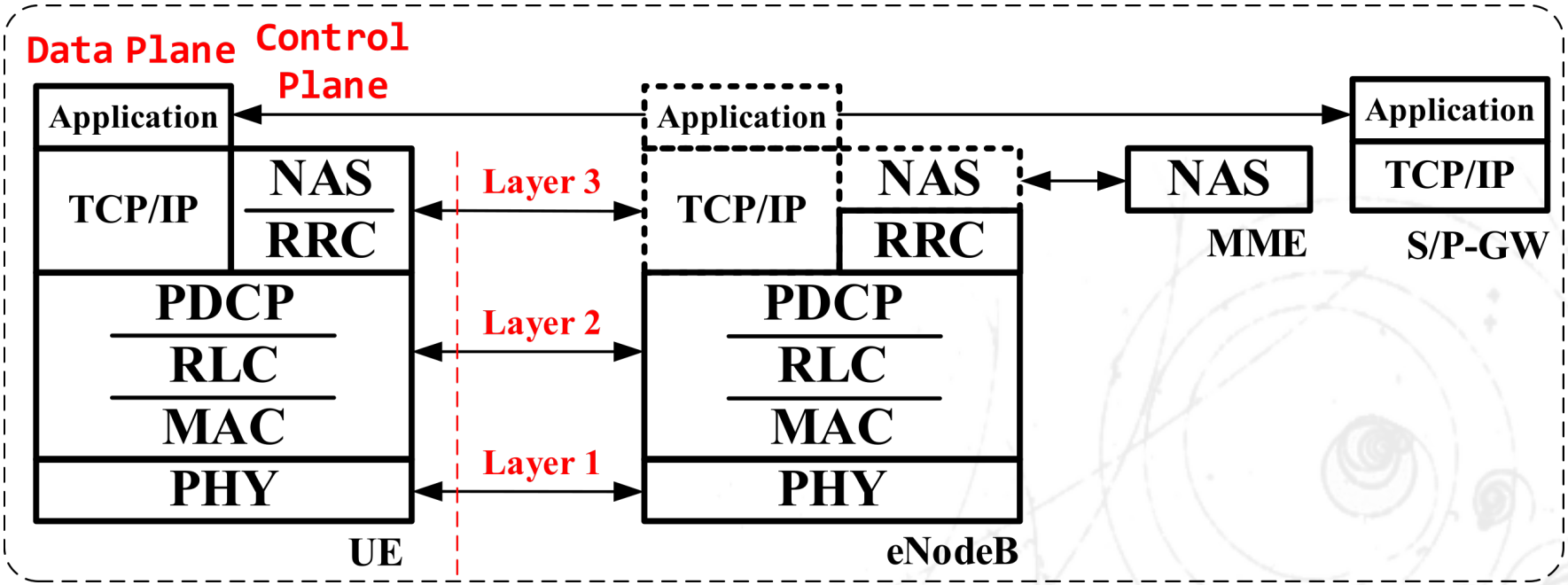
User Identifiers in LTE Networks

- **Importance of Identifiers:** Ensuring Proper Data Delivery and Secure Transmission
- **Radio Network Temporary Identifier (RNTI):**
  - Assignment: By the eNB to the Connected UE
  - Differentiation: Distinguishing the User from Other Connected UEs
  - Validity: Only During UE's Connection to the Serving eNB
- **Temporary Mobile Subscriber Identity (TMSI):**
  - Assignment: By the EPC (Evolved Packet Core) in the LTE Network
  - Uniqueness: Newly Assigned upon UE Registration

Background

RAN in LTE

- **Radio Access Network (RAN):** Network Between UE and eNB
- **RAN Function:** Transmission of Control Messages and User Data Traffic
- **LTE Protocol Stack:**



Background

Encapsulation and Delivery of Control Messages and User Data Traffic

- **Packet Data Convergence Protocol (PDCP) Layer:**
  Encryption of Control Messages and User Data Traffic
- **Radio Link Control (RLC) Layer:**
  Segmentation of Control Messages and User Data Traffic
- **Medium Access Control (MAC) Layer:**
  Scheduling of Control Messages and User Data Traffic
- **Physical Layer (PHY):**
  Delivery of Control Messages and User Data Traffic

Background

RRC Protocol in LTE

- **Radio Resource Control (RRC) Protocol:** Managing Radio Connections
- **RRC Connection Setup:**
  1. UE Request for New Connection to eNB
  2. eNB Responds with RRC Connection Setup Message
  3. Delivery of Wireless Configuration Information
- **RRC Connection Release:**
  1. No Traffic between eNB and UE for Specified Period
  2. eNB Sends RRC Connection Release Message
  3. Removal of Connection to Save Network Resources and UE's Battery

Background

Data Acquisition in LTE

- **Channel Broadcasting:**
  - Signals Broadcast to Devices Over Range of Radio Frequencies(Channel)
- **Selective Retrieval of Data:** Each UE Retrieves Own Data from Broadcast Signals
- **Physical Layer Instructions:**
  - **Retrieving Control Information:** Physical Downlink Control Channel (PDCCH)
  - **Retrieving User Data:** Physical Downlink Shared Channel (PDSCH)

Background

DCI and PDCCH in LTE

- **PDCCH:**
  - Unencrypted Channel!
- **DCI Transmission**:
  - Downlink Control Information (DCIs) are Transmitted Over PDCCH Channels
  - DCIs Transmitted on Every LTE Subframe (1 ms)
  - Exposure of DCIs of All LTE Users on the Same Carrier Frequency (Thanks to PDCCH)
- **DCI Components:**
  - **Assigned Resource Blocks:** Frequency and Timing for Data Retrieval
  - **Modulation Coding Scheme:** Decoding Scheme for Assigned Blocks
  - **CRC Bits with RNTI:** Error Detection with RNTI
- Single UE Capable of Decoding Every DCI on the Same PDCCH

Background

Security and Privacy in LTE Data Transmission

- **Data Retrieval in LTE:** UEs Sharing the Same eNB Retrieve All Other User's Data from PDSCH
- **Secrecy and Integrity Protection:** Ensuring Legitimate User Access to Own Data
- **Encryption and Integrity Checking:** Provided at the PDCP Layer in LTE Networks
- **Unencrypted Protocol Layer Headers:** Headers of Layers Underneath IP Layer Not Encrypted
- **Exploiting Unencrypted Headers:** Approximating Volume of Video Streaming Traffic

Background

Carrier Aggregation in LTE Networks

- **Enhancing Bandwidth with CA:**
  - Simultaneous Use of Multiple Carrier Frequencies
  - Providing More Bandwidth per User
- **eNB and Cell Structure:** Multiple Cells within an eNB, Each Covering a Frequency Band
  - Primary Cell (PCell):
  - Secondary Cells (SCells):
- **Managing CA at the MAC Layer:** Multiplexing and Delivering Packets Over PCell and SCells

Video Identification Attack

Introduction

- **Attack Objective:** Precisely Inferring the Video Playing on a Victim's Device
- **Targeted Network Traffic:** Encrypted Traffic Sent to the Victim's Device Through LTE Network
- **Exploiting Operational Logic of HAS:**
  - Information Leakage from HTTP Adaptive Streaming (HAS)
  - Dominant Streaming Protocol Used by Popular Services

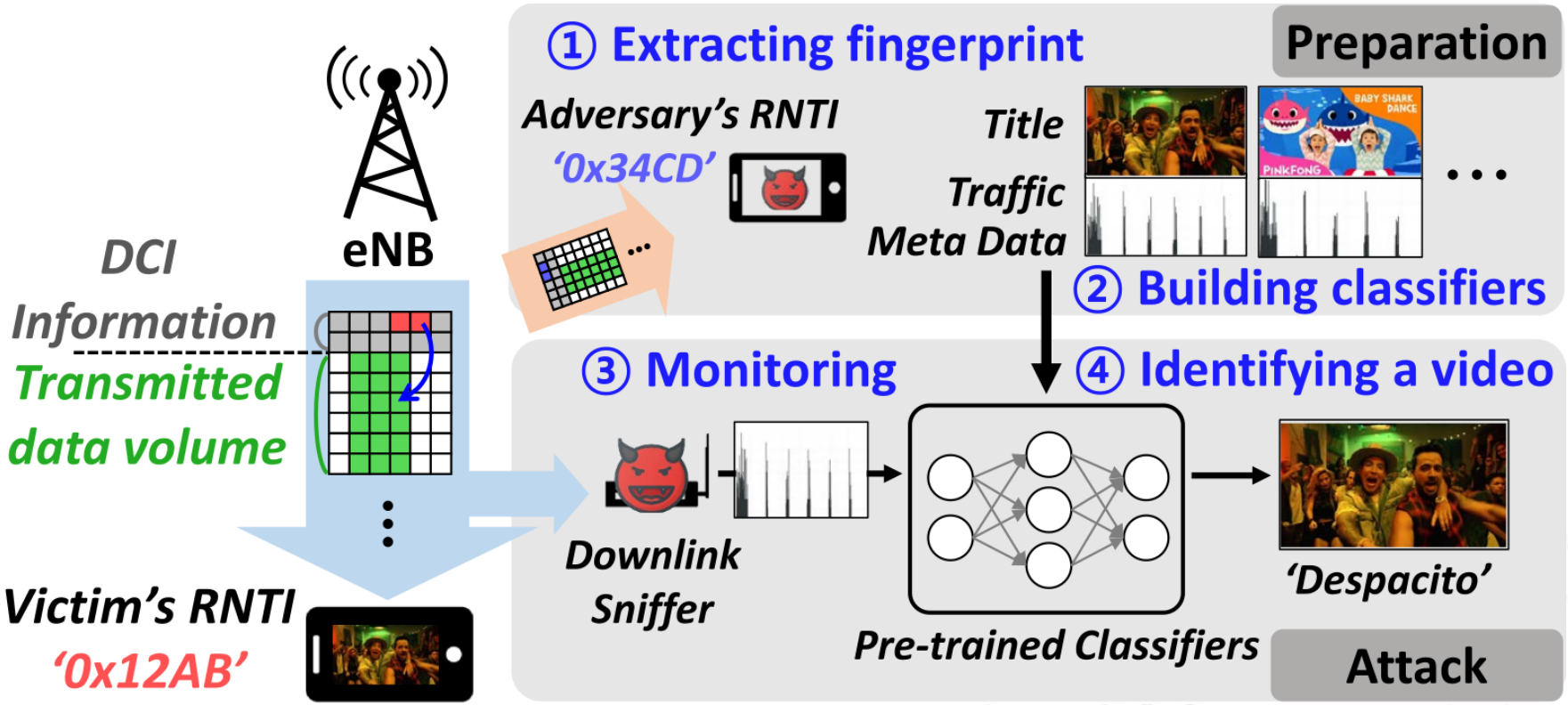
Video Identification Attack

HTTP Adaptive Streaming

- **HAS Chunk Segmentation:**
  - Video Divided into Smaller Chunks
  - Chunks Share Identical Playback Time
- **Variable Bitrate Encoding and Chunk Sizes:**
  - Different Sizes for Chunks Based on Content
  - VBR Encoding
- **Unique Pattern Formation:**
  - Series of Chunks Representing a Distinct Pattern
  - Unique Pattern Based on Chunk Sizes
- **Video Fetching and ON/OFF Periods:**
  - HAS Client Fetches Multiple Chunks Periodically

Video Identification Attack

Attack Overview



Video Identification Attack

Video Identification Process (I)

1. **Recording**
2. **Building a classifier**
3. **Monitoring**
4. **Identification**

Video Identification Attack

Video Identification Process (II)

- **Recording:**
  - Adversary selects target videos for identification
  - Generates network traffic by playing the videos repeatedly
  - Extracts distinctive features from the observed traffic
  - Prior studies have considered identifiable features such as:
    - Number of downlink and uplink packets
    - Volume of downlink and uplink packets
    - Arrival interval of downlink and uplink packets

Video Identification Attack

Video Identification Process (III)

- **Building a classifier:**
  - Adversary constructs a classifier to identify videos
  - Prior research has used various techniques:
    - CNN
    - Support Vector Machine (SVM)
    - K-nearest Neighbors with Dynamic Time Warping (DTW)

Video Identification Attack

Threat Model

- **Attacker's Goal**
- **Attacker's Capabilities**
- **Implication**

Video Identification Attack

Threat Model: Attacker's Goal

- Adversary selects a target cell and set of target videos
- **Goal:**
  Recognize identifiers (TMSI and RNTIs) of UEs in the cell who watched target videos

Summary

For each identified UE, the adversary aims to identify the specific video title watched

Video Identification Attack

Threat Model: Attacker's Capabilities (I)

- **No direct access** to victims' UEs or the eNB
- Lack of information about victims:
  - Usage Details
  - Cryptographic Keys
- Inability to decrypt user data in PDCP packets

Video Identification Attack

Threat Model: Attacker's Capabilities (II)

- Can choose any target eNB for the attack
- **Eavesdrops on radio signals transmitted over the air:** Utilizes a SDR and publicly available software

- Parses downlink messages to obtain MAC packets, DCIs, and unencrypted PDCP header information

Video Identification Attack

Threat Model: Implication

Implication

Adversary can determine specific videos being streamed solely by observing **encrypted** video traces

Tracking Users of Illegal or Sensitive Videos

- Scenario:
  Law enforcement agency in an oppressive regime executes the attack
- Personal identities linked to identifiers with ISP assistance or further identifier linking attacks

Video Identification Attack

Practical Challenges: Limited monitoring capability (I)

- Adversary lacks the ability to **decrypt** encrypted user traffic over the PDSCH
- **Mimicking** procedure decodes DCIs from broadcast signals
- Obtains time series of **packet volumes** as a pattern for each video

Video Identification Attack

Practical Challenges: Limited monitoring capability (II)

- **Inconsistent** size information from extracted DCIs hinders accurate video identification
- Computed packet volumes include **non-video traffic**, impacting identification precision
  - Retransmitted packets at RAN layers (RLC, PDCP) due to network conditions
  - LTE control plane messages that add to the traffic volume
  - Presence of packets from other applications running on the victim's UE
- Occasional decoding failures of the downlink sniffer further obscure chunk size

Video Identification Attack

Practical Challenges: Limited monitoring capability (III)

- Comparison between estimated video size (decoded DCIs) and actual size at the IP layer
  - Range of difference: -1.1% to 9.6% with an average of 7%
- Adversary must develop a robust classifier to account for inconsistent volumes
  - **No Worries! Use Machine Learning!**

Video Identification Attack

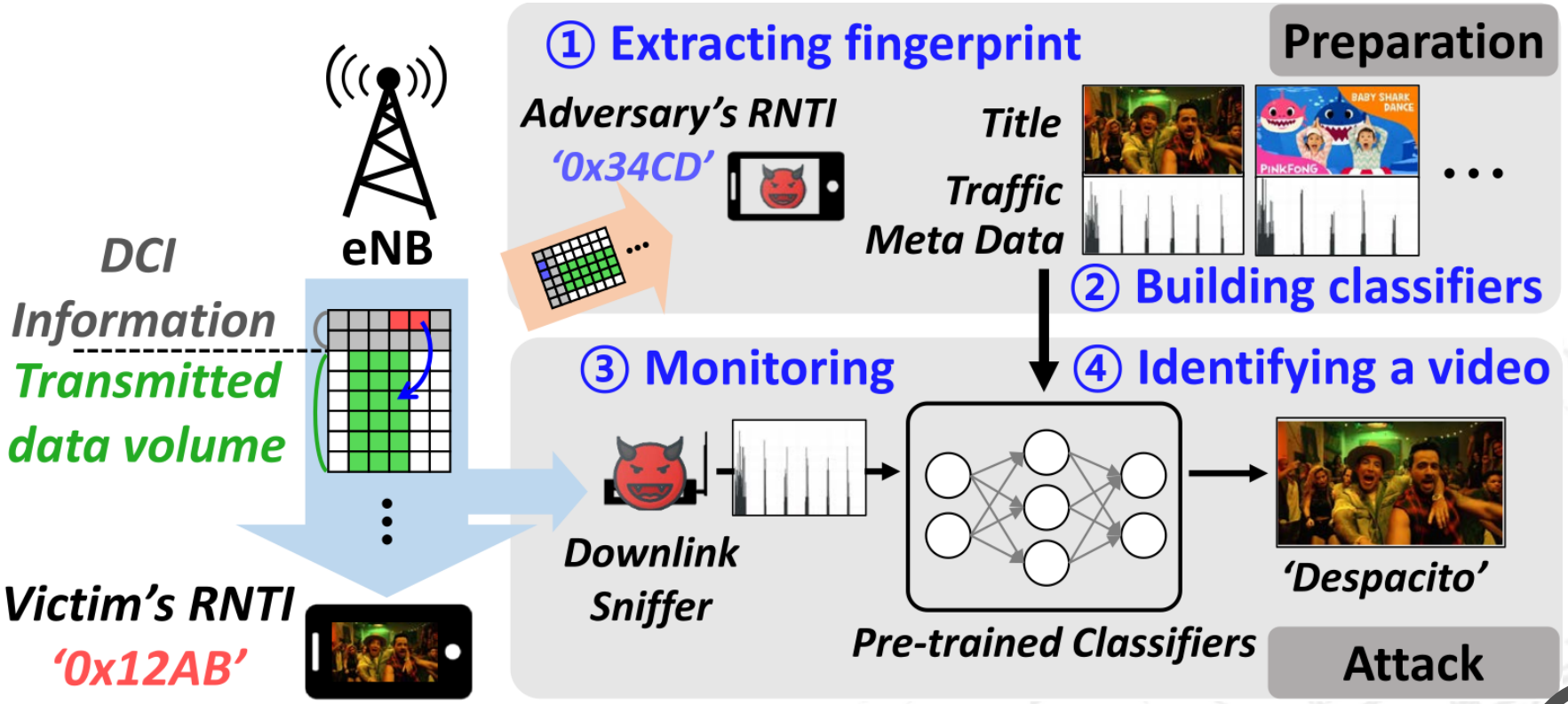
Practical Challenges: Limited monitoring capability (IV)

- Video service providers significantly impact streaming traffic patterns
- Service providers affect traffic volume estimation in **CA-enabled** environment
- Adversary cannot utilize known IP addresses due to encryption at TCP/IP layer!
  - **No Worries! Use Machine Learning**

Video Identification Attack

Attack Review

- The attacker selects target videos and computes their fingerprints
- The attacker prepares the attack by building a classifier that identifies the target videos



Video Identification Attack

Attack: Adversary's Initial Data Collection

- **Objective:**
  Obtain time series of transmitted data volumes for target videos
- Adversary plays target videos **repeatedly** on their own UE
- Adversary decodes DCIs to obtain a **time series** of transmitted **data volumes**
- Adversary retrieves their own RNTI from diagnostic monitoring tools connected on their UE (Cellular-Z)
- Adversary configures their UE to use only **one band**, disabling CA (C2 and C3)

Video Identification Attack

Attack: Adversary's Classifier

- **Objective:**
  Build a robust classifier
- **Challenges:** Inconsistent input traces from the same video
  - Using **CNN** As a Classifier!
    Robust against unsteady input patterns with noise
    - Convolution Layer:
      Shared weights and parameter sharing
    - Pooling layers
    - Dropout

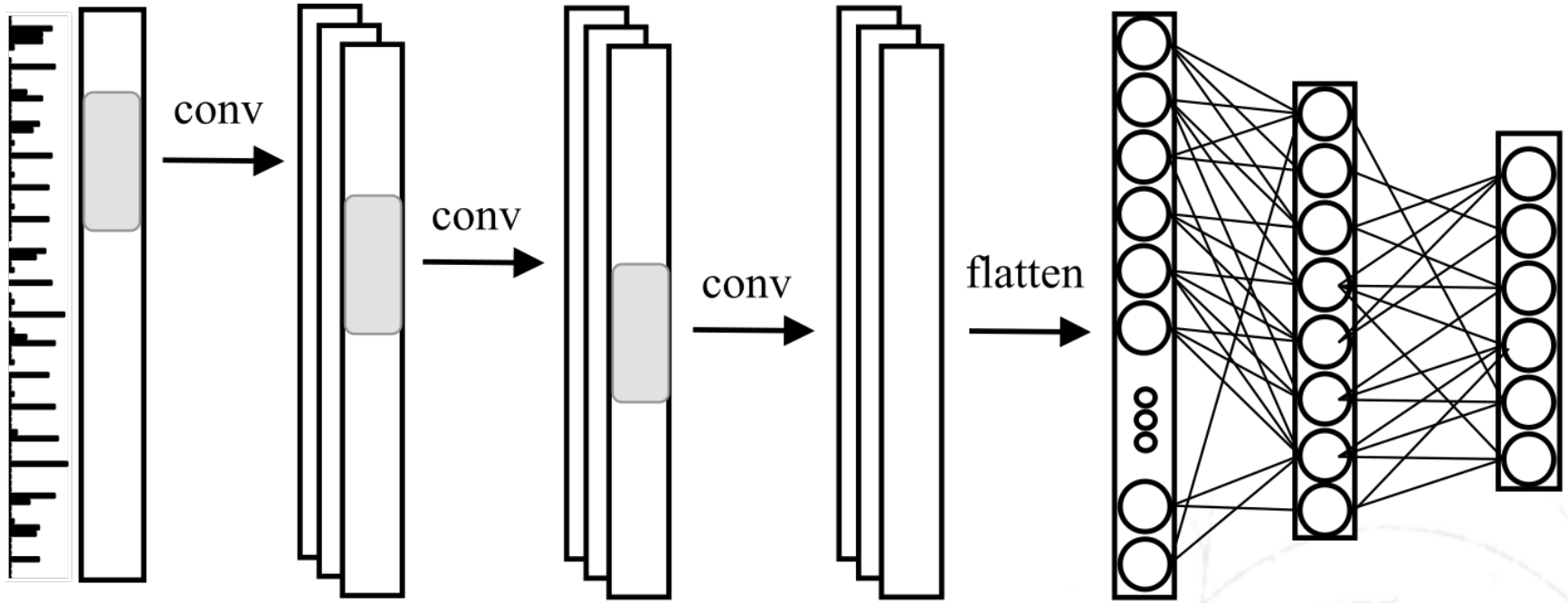
Video Identification Attack

Attack: Accurate Video Identification

- **Objective:**
  Infer the video service provider based on operational logic differences in HAS implementations by various vendors
  - Decision Tree Classifier
- Train **multiple CNN classifiers**, each specialized for a specific video service provider

Video Identification Attack

Building a Classifier: Architecture (I)



Video Identification Attack

Building a Classifier: Architecture (II)

- **Input**:
  Vector representation of the time series of estimated traffic volumes
- Traffic volume for 0.2s (i.e., 200 LTE subframes)

Video Identification Attack

Building a Classifier: Architecture (III)

- **Three convolution layers**
  - Each 150 Filters (Empirically): Covering 20 vector elements
  - Each Accompanies a **Maxpool** and **Dropout**
- **ReLU** Activation Function
- **Dropout Rate** of 0.4
- Pool Size of 3
- Two **Dense Layer**
  - Layer 1: 500 Units
  - Layer 2: Number of Video Titles (Obviously!)
- 50 Epochs, Adam Optimizer
- Batch Size of 32
- Other HyperParameters: Default (**Keras**)

Video Identification Attack

Building a Classifier: Dataset (I)

- Sniffing LTE signals from **three** major MNOs with commercial UEs
- **2,035 hours** of streaming time and **1.79 TB** of video traffic

Dataset	# of Videos	# of Traces	Trace Length (s)	Description
YouTube100	100	29,715	120	YouTube Top 100
Netflix	22	1,001	800	Netflix Top 50
NetflixDT	31	298	800	Netflix (Random)
Amazon	32	1,210	120	Prime Video (Random)
AmazonDT	36	310	120	Prime Video (Random)
YouTubeCA	100	7,383	120	YouTube Top 100 (CA)
YouTube200	200	6,424	120	YouTube Top 101-300
Web	-	268	120	Alexa Top 50
Teleconf	-	201	120	Google Meet

Video Identification Attack

Building a Classifier: Dataset (II)

- **Target videos:**
  Most viewed music videos on YouTube (Top 100) with lengths under 6 min each
  - Premium Account (No Ads!)
  - Multiple Traces For Each Video
    - Played each video over 27 times
- 480p, 720p, 1080p
- Netflix + Amazon (Out of Scope)
- Different MNOs (Out of Scope)

Video Identification Attack

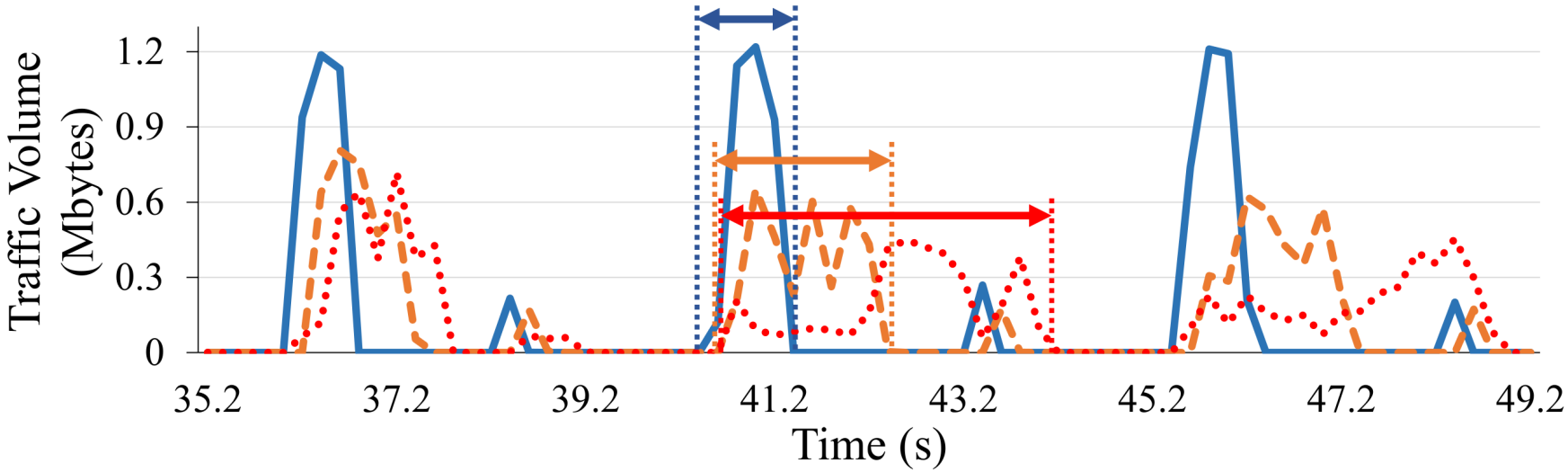
Evaluation: YouTube100

- Results are Pretty good
  - In Which Scenario?
- YouTube100 Dataset
- Different MNOs, Quality, Mixed Quality

MNO	Video Quality	# of Traces	Acc.	F_1	AUC	MNO	Video Quality	# of Traces	Acc.	F_1	AUC
A	480p	3,184	0.970	0.967	0.997	C	480p	3,262	0.981	0.961	0.996
A	720p	3,196	0.968	0.965	0.998	C	720p	3,218	0.967	0.965	0.997
A	1080p	3,645	0.985	0.986	0.998	C	1080p	3,293	0.957	0.954	0.997
B	480p	3,376	0.928	0.922	0.995	A	mixed	10,025	0.977	0.976	0.998
B	720p	3,318	0.973	0.970	0.997	B	mixed	9,917	0.958	0.958	0.997
B	1080p	3,223	0.940	0.937	0.996	C	mixed	9,773	0.980	0.979	0.998

Video Identification Attack

Evaluation: Impact of Network Environment



Video Identification Attack

Evaluation: Results on Youtube100

MNO	Video Quality	# of Traces	Acc.	F_1
A	480p	3,184	0.970	0.967
A	720p	3,196	0.968	0.965
A	1080p	3,645	0.985	0.986

Video Identification Attack

Evaluation: Metrics

- Consider a **Binary** Classification
- **Confusion Matrix**
- F1: Balanced classification

		Predicted Failure: True	Predicted Failure: False
Actual Failure	True	TP := True Positive	FN := False Negative
	False	FP := False Positive	TN := True Negative

→ *Recall := TP / (TP+FN)*

Precision := TP / (TP+FP) → *F1-Score := 2*Precision*Recall / (Precision+Recall)*

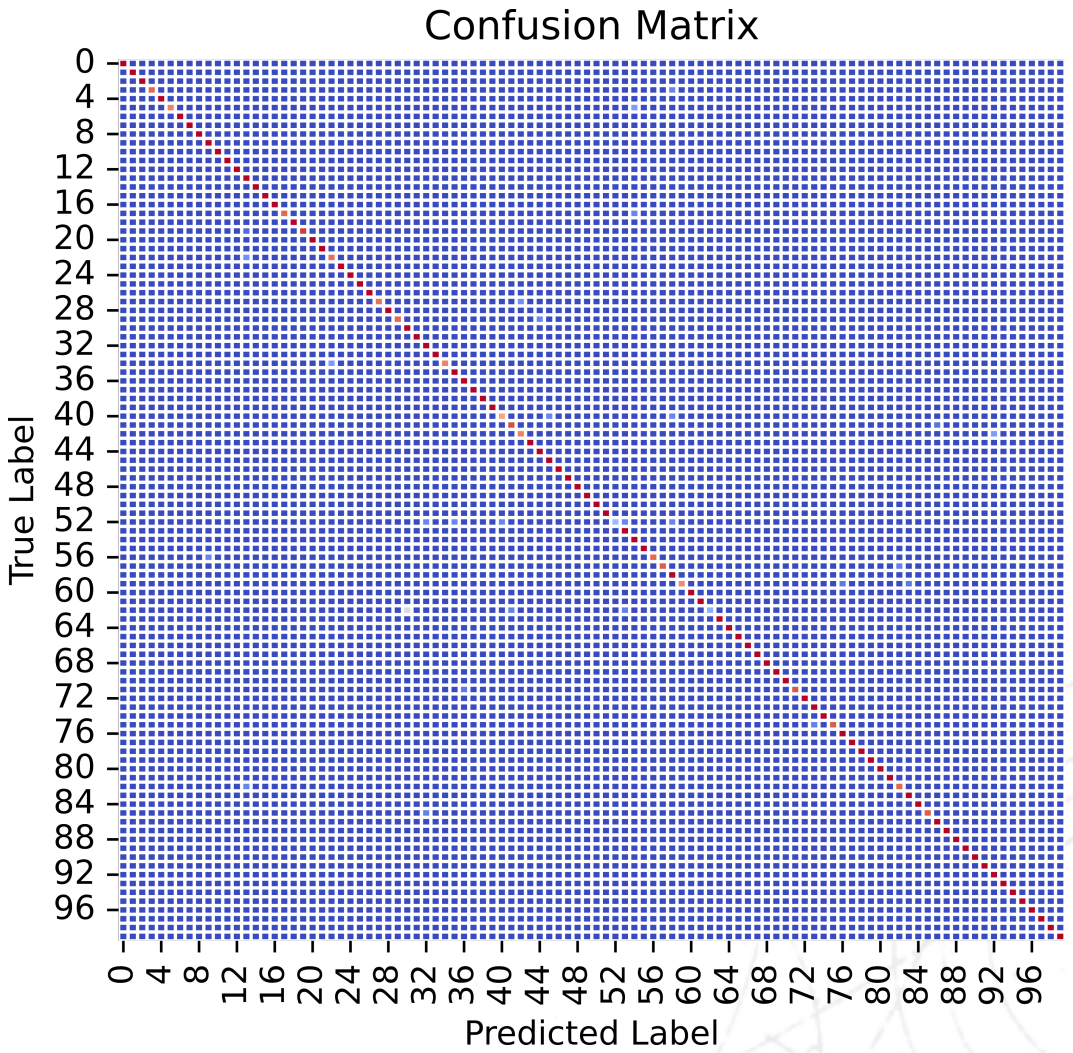
Video Identification Attack

Evaluation: Results (I)-A

- Architecture
- 10 Epochs!
- Epoch 10/10: Loss: 0.0196 - Accuracy: 0.9949
- Accuracy on test data: 0.95571
- F1-Score: 0.96

Video Identification Attack

Evaluation: Results (I)-B



Video Identification Attack

Evaluation: Results (II)

- Same Training Set, Testing Set, Architecture
- 30 Epochs!
- Epoch 10/10: Loss: 0.0108 - Accuracy: 0.9969
- Accuracy on test data: 0.96857
- F1-Score: 0.97

Video Identification Attack

Evaluation: Results (III)

- Same Training Set, Testing Set
  Three convolution layers
  - Each 150 Filters (Empirically)
- 30 Epochs!
- Epoch 10/10: Loss: 4.6661e-05 - Accuracy: 1.0000
- Accuracy on test data: 0.98143
- F1-Score: 0.9816

Video Identification Attack

Evaluation: Results (IV)

- Let's Change Training Set, Testing Set
- **Three convolution layers**
  - Each 150 Filters (Empirically)
- 30 Epochs!
- Epoch 10/10: Loss: 6.7214e-06 - Accuracy: 1.0000
- Accuracy on test data: 0.81884
- F1-Score: 0.82

Video Identification Attack

Evaluation: Better Results

- Filter Size!
- Pooling Size!
- Better Architecture

Video Identification Attack

Evaluation: Scalability

Class Size	50	100	150	200	250	300
Acc.	0.994	0.986	0.985	0.980	0.978	0.978
F_1	0.993	0.985	0.985	0.979	0.980	0.977