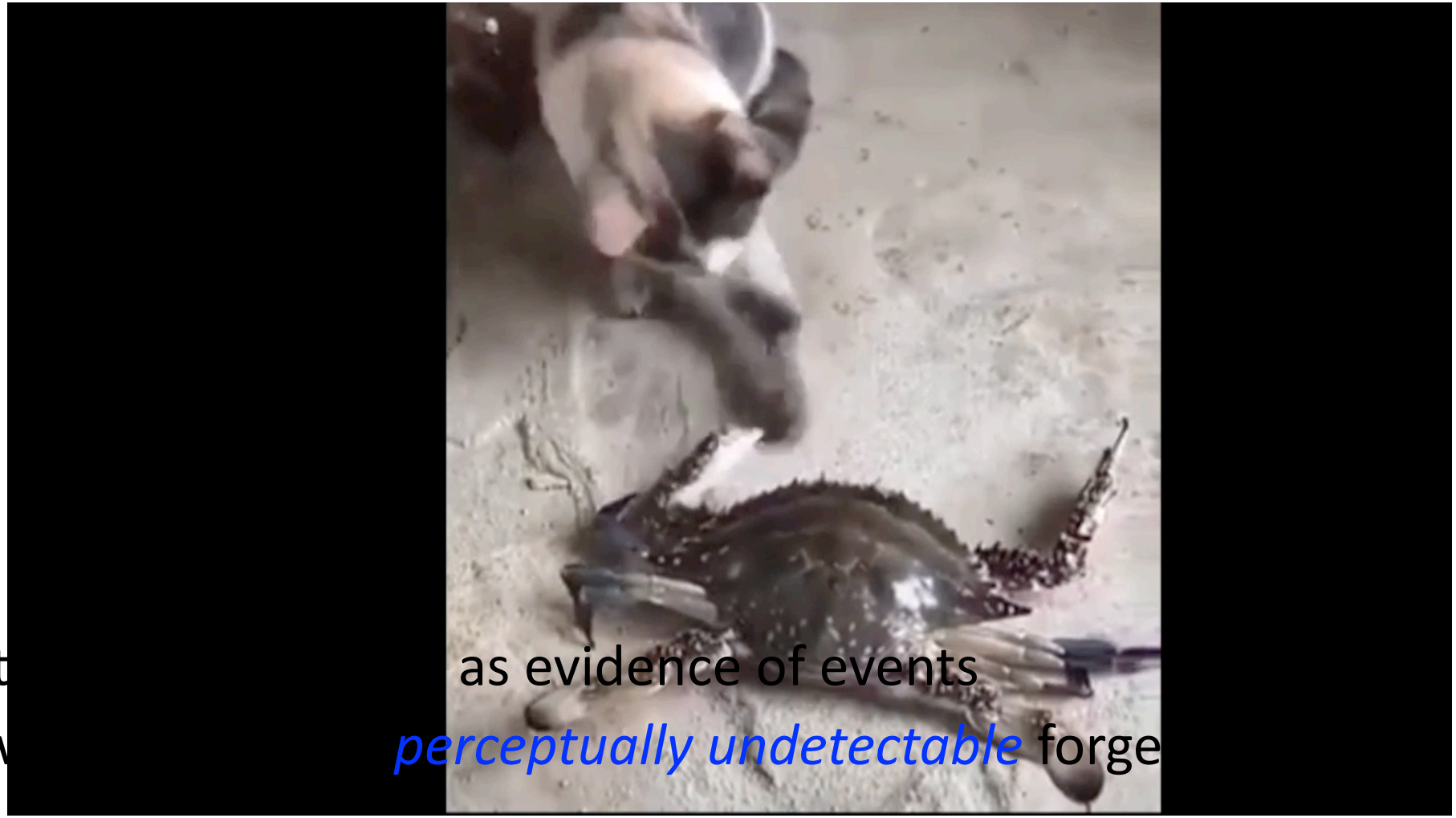
# A New Approach to Detecting Frame Deletion in H.264 Encoded Digital Video

***Hunter Kippen***
Advisor: Dr. Matthew Stamm
Drexel University
hmk64@drexel.edu


# Video Tampering



- Digit
- Softv

as evidence of events

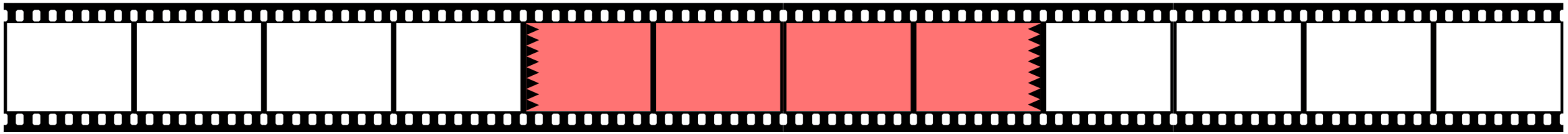
*perceptually undetectable* forge

# Multimedia Forensics

- Forensics can provide information about authenticity when the *information source is not trusted*
- Detect traces known as *fingerprints* left by processing and manipulation
  - Left by the capture process
  - Introduced by editing operations
- Detecting fingerprints left by editing is known as *forgery detection*

# Video Frame Deletion

- Frame deletion occurs when a *frame* or *sequence of frames* are removed from a video
- Potential effects include:
  - Removing the *context*
  - Removing the presence of objects
- *Often visually undetectable*



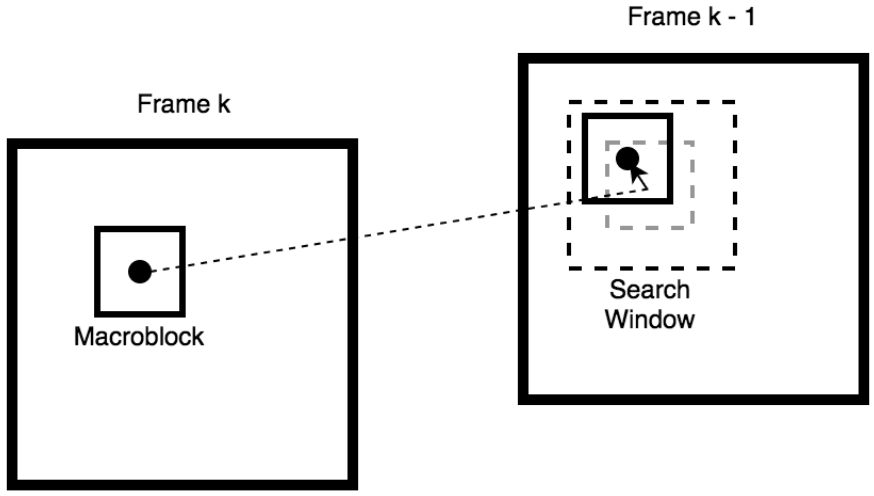
# Video Compression Overview

- Subsequent video frames are similar
- Can *predict* the content of the current frame from a previous frame


Second Frame | First Frame | Error Residual

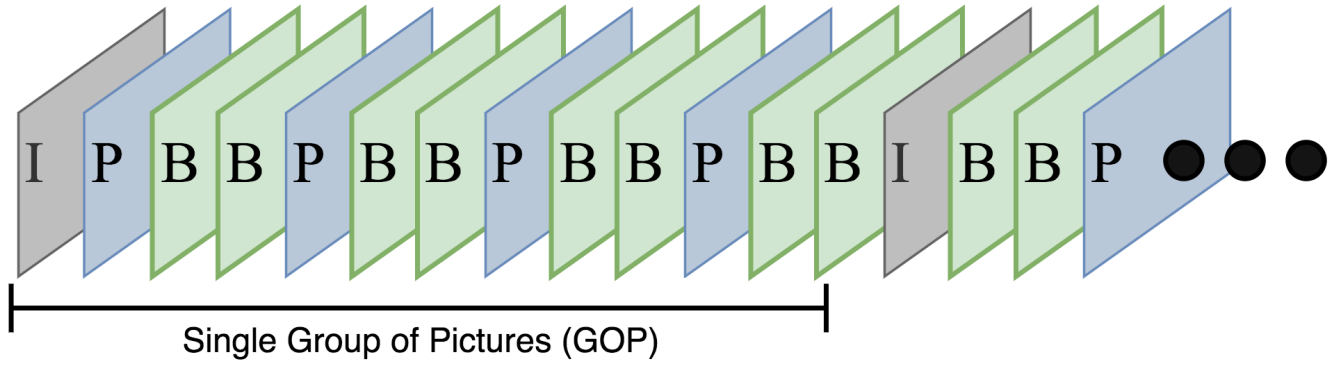
# Video Compression Overview

- Frames are partitioned into *macroblocks*
- Encoder finds *least error* mapping into reference frame
  - Block Matching
- X and Y displacement for mapping recorded
- Encoder stores *motion vectors* and *error residuals*



# Group of Pictures

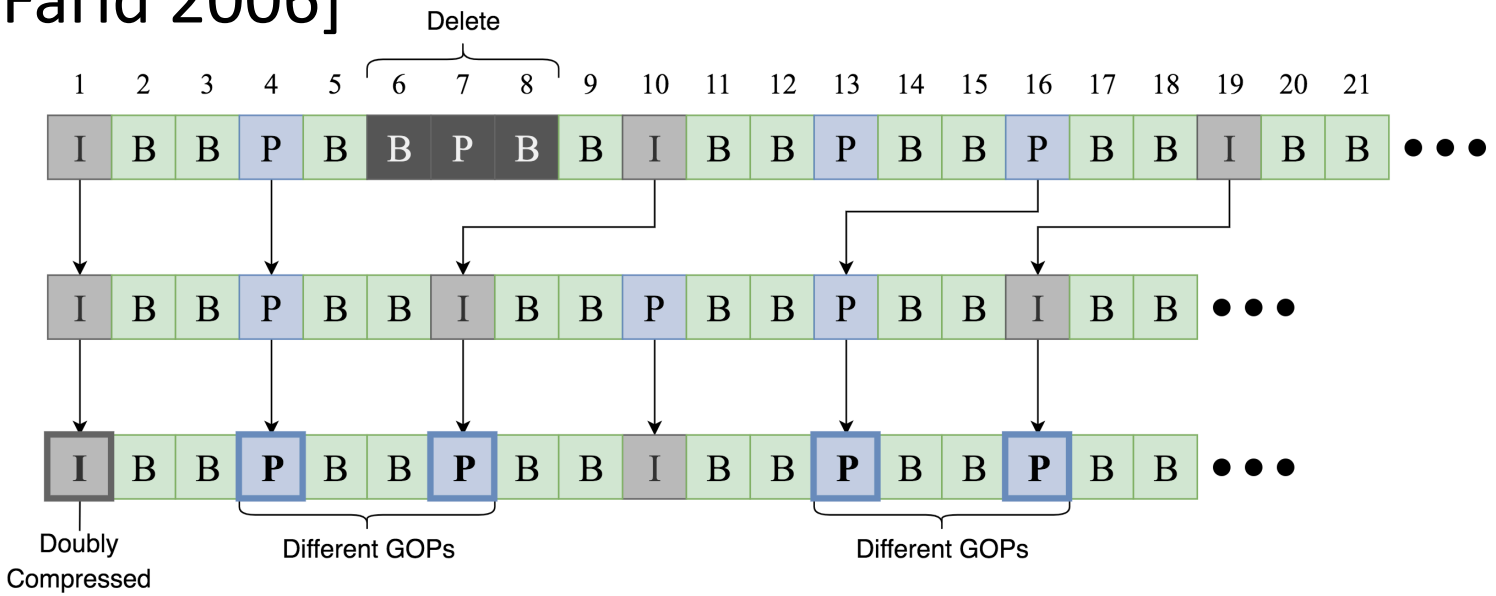
- Frames are separated into *Groups of Pictures* (GOP)
  - Prediction occurs only *within* each GOP




- Each GOP contains three frame types
  - Intra-frames (I-frames)
  - Predicted-frames (P-frames)
  - Bidirectional-frames (B-frames)

# Frame Deletion Fingerprints

- Frame deletion affects P-frame prediction error
  - Frames predicted within same GOP have less prediction error
  - Frames predicted *across* old GOPs have *greater* prediction error
- Frame deletion and addition introduces a *temporal fingerprint* [Wang & Farid 2006]



# Frame Deletion Fingerprints

- Define P-frame error signal as $e(n) = \frac{1}{N_{xy}} \sum_x \sum_y |p_{x,y}(n)|$

P-frame index

prediction error

- Temporal fingerprint results in a pattern of *increased* $e(n)$ values
- Can *visually inspect* the DFT of the P-frame error signal to detect frame deletion [Wang & Farid 2006]
- Visual inspection is prone to human error
  - Not feasible for large scale operation

# Automatic Detection

- Automatic detection methods were formulated for detecting frame deletion [Stamm et al. 2012]
- Model prediction error sequence for video with frame deletion

$$e_2(n) = e_1\left(n - n_D\right)\left(1 + s(n)\right)$$

Temporal fingerprint

- Detection formulated as a hypothesis test

$$\begin{cases} H_0 : e(n) = e_1(n) \\ H_1 : e(n) = e_2(n) = e_1(n) + e_1(n)s(n) \end{cases}$$

# Automatic Detection

- Need to extract estimate of temporal fingerprint
- Process incoming $e(n)$ to estimate $e_1(n)$

$$\hat{e}(n) = \text{median}\{e(n-1), e(n), e(n+1)\}$$

$$\implies e_1(n) = \hat{e}(n) + \epsilon(n)$$

- From $\hat{e}(n)$ estimate $s(n)$

$$\hat{s}(n) = \max(e(n) - \hat{e}(n), 0)$$

# Automatic Detection

- Reframe hypothesis test in terms of $\hat{s}(n)$

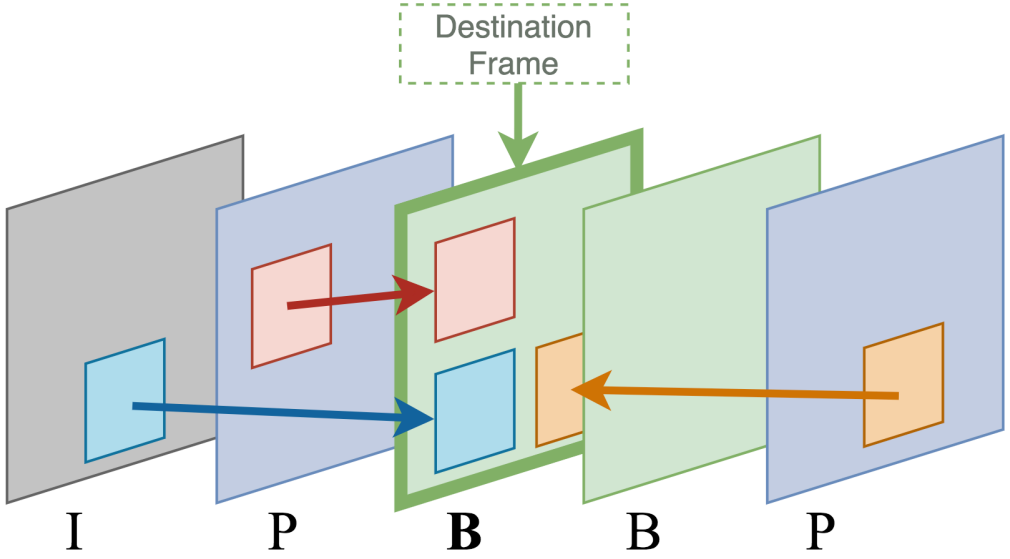
$$\begin{cases} H_0 : \hat{s}(n) = \max(\epsilon(n), 0) \\ H_1 : \hat{s}(n) = \max(e(n)s(n) + \epsilon(n), 0) \end{cases}$$

- Use a decision function based on the energy of $\hat{s}(n)$ [Stamm et al. 2012]

$$\delta_{var} = \begin{cases} H_0 : \text{if } \frac{1}{N}\sum_{n=1}^{N} |\hat{s}(n)| < \tau_{var} \\ H_1 : \text{if } \frac{1}{N}\sum_{n=1}^{N} |\hat{s}(n)| \geq \tau_{var} \end{cases}$$

# Automatic Detection

- Decision function formulated for *MPEG-2* video
- H.264 standard *updates* inter-frame prediction process




- It is *unclear* whether the frame deletion fingerprint is expressed in H.264 encoded video.

# Preliminary Experiment

- Testing detector performance on H.264 video requires a dataset
- The dataset was gathered using one camera model
  - ASUS ZenFone 3 Laser
  - 257 - 5 second long videos
  - Variety of lighting conditions (indoor & outdoor)
  - Variety of motion and scene content
- For each video in the dataset, we
  - Decoded the video
  - Removed the first 15 frames (Half GOP)
  - Reencoded the video using the same encoding parameters

# Preliminary Experiment

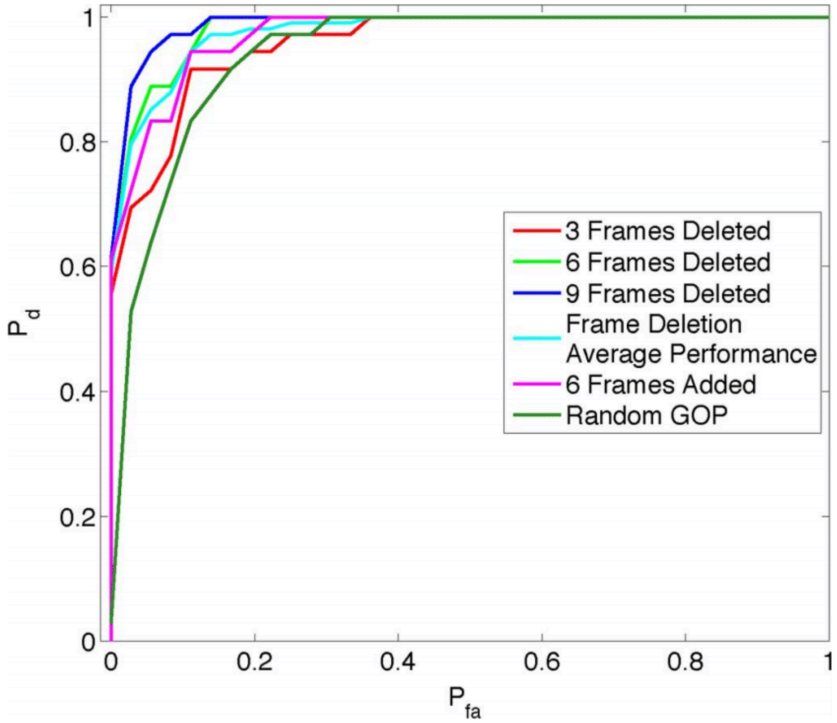
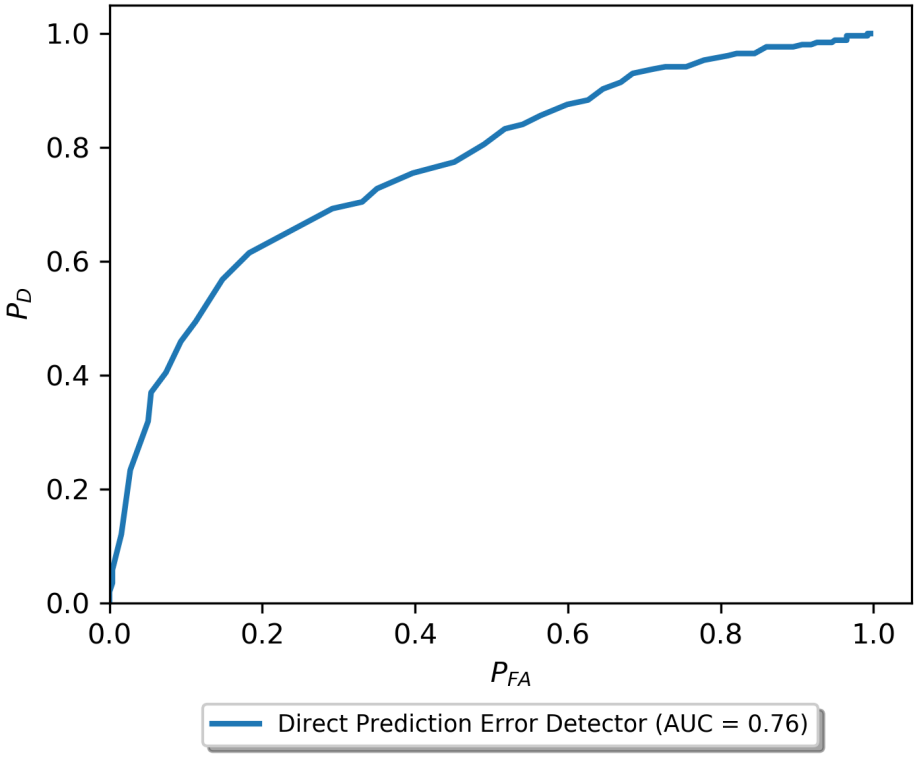
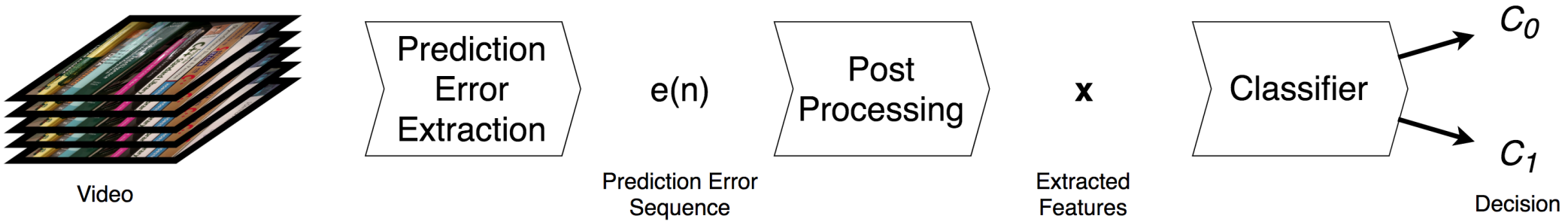



*Figure reprinted with permission from Stamm et al.




- Fingerprint is *not strongly expressed* in H.264 video
- We need new methods to isolate the fingerprint

# General Frame Deletion Detection Approach

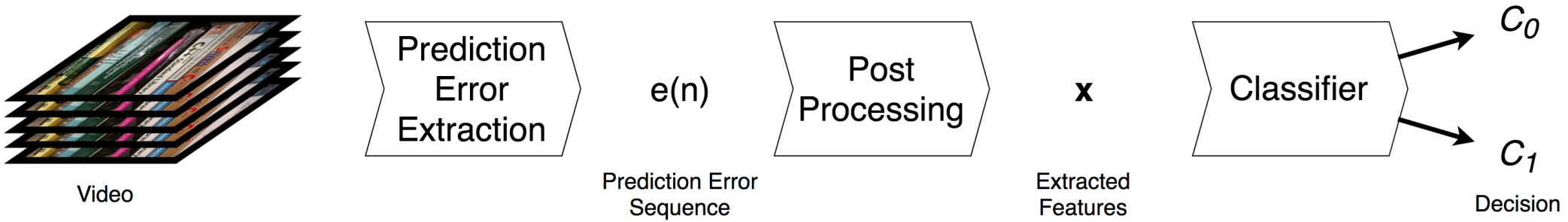



- Frame deletion detection can be thought of as a *binary classification problem*
- Given a video *V*

$C_0$ : The video is *genuine*, and has not had frames removed from it.

$C_1$ : The video is *altered*, and has had frames removed from it.

# General Frame Deletion Detection Approach




- Given extracted features **x**

$C_0 : \boldsymbol{x}$ resulted from a *genuine* video that has not had frames removed from it.

$C_1 : \boldsymbol{x}$ resulted from an *altered* video which has had frames removed from it.

# Assumptions

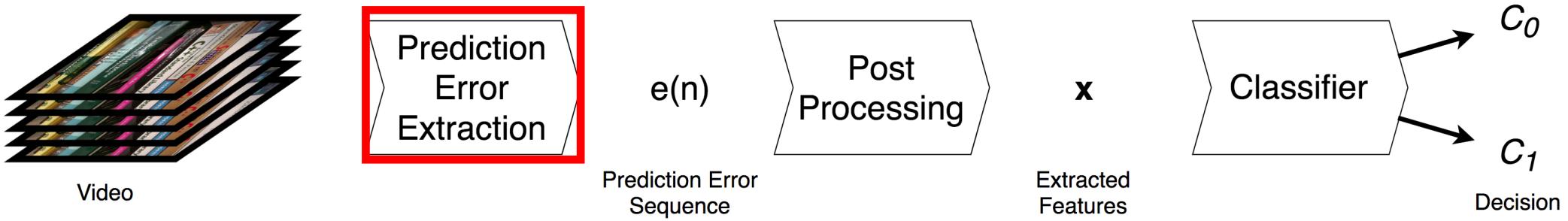
- All source videos have been initially compressed using H.264 or MPEG-4
- Altered video undergoes *recompression* using H.264 *after* frame deletion
- GOP structure of reencoded video matches original video
- Videos are of *sufficient length*
- Frame deletion does not occur at the *end* of the video
- The number of frames deleted is not a multiple of a full GOP

# Assumptions

- A user will not need physical access to devices
- Function on videos of any
  - Length
  - Resolution
  - Frame rate
  - GOP structure

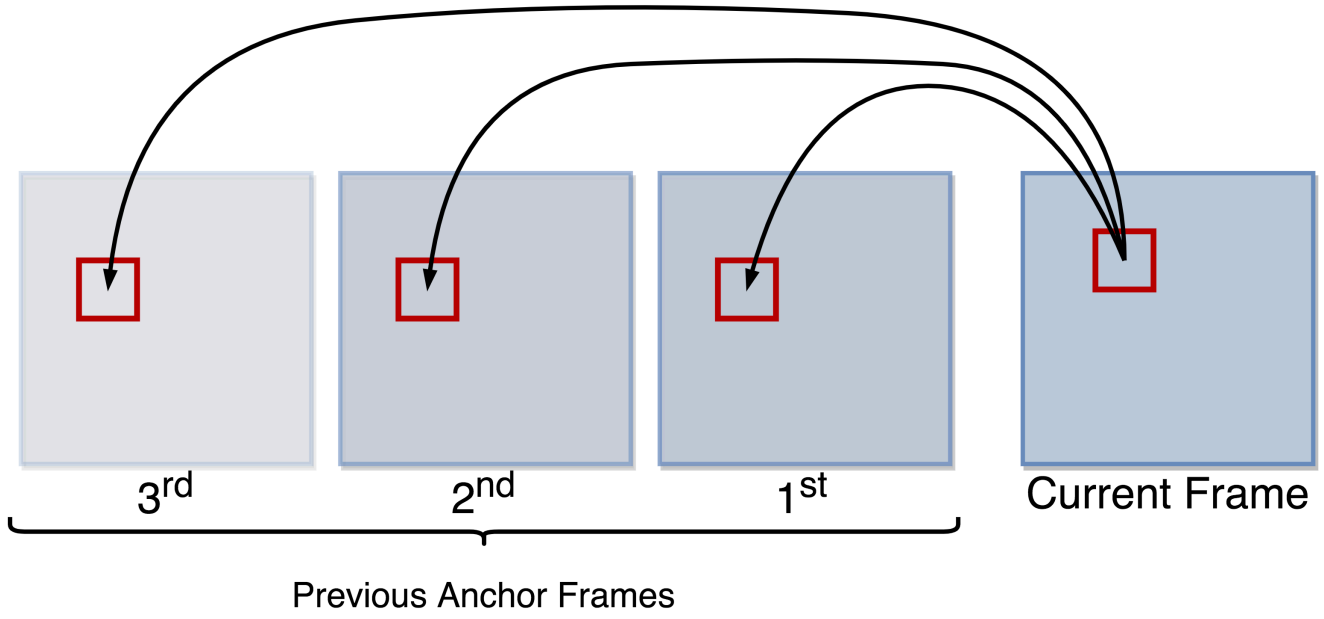
- *Data driven approach*
  - Labeled Dataset required

# Proposed Prediction Error Extraction Method



# Proposed Prediction Error Extraction Method

- Our proposed method should *maximize* the probability of observing a cross-GOP prediction
- Instead of directly observing P-frame prediction error



# Proposed Prediction Error Extraction Method

- Mean *absolute* error of each mapping stored as a matrix $M(n)$

$$M_{i,j}(n) = \frac{1}{N_i} \sum_x \sum_y |p_{i,j}(x, y, n)|$$

Motion vector index — $j^{th}$ Previous anchor frame

- Keep only the *minimum* value of each row in $\tilde{M}(n)$

$$\tilde{M}_{i,j}(n) = \mathbb{1}\left(j = \underset{l}{\operatorname{argmin}}\left(M_{i,l}(n)\right)\right) * M_{i,j}(n)$$

# Proposed Prediction Error Extraction Method

- Determine *average error* for each previous anchor frame

$$P_j(n) = \frac{1}{N_j} \sum_i \tilde{M}_{i,j}(n)$$

- Report the *maximum* value of $P(n)$

$$e^*(n) = \max_j P_j(n)$$

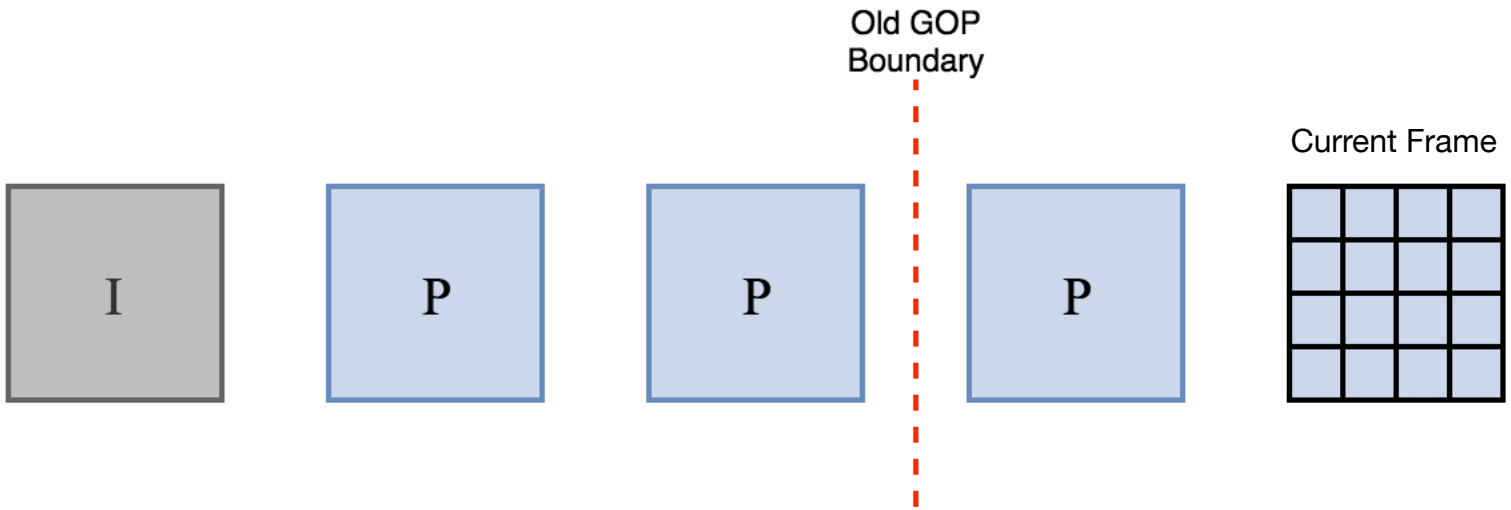
# Proposed Prediction Error Extraction Method

$$M[n] = \begin{bmatrix} 30.4 & 20.5 & 40.2 \\ 16.3 & 22.1 & 30.4 \\ 25.5 & 23.4 & 19.8 \\ \vdots & \vdots & \vdots \end{bmatrix}$$

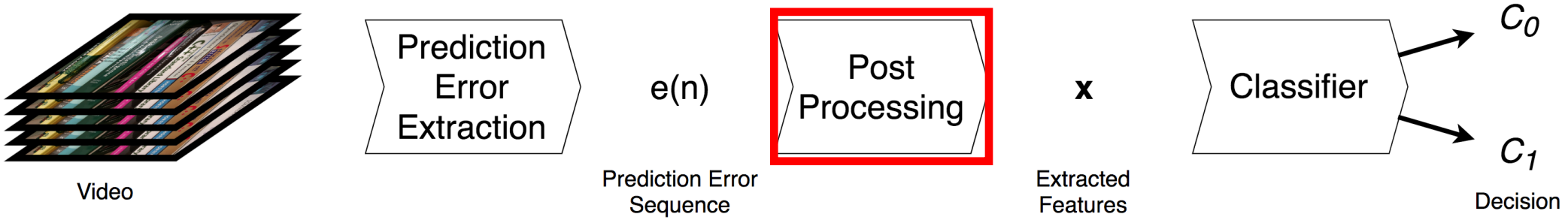
$$\tilde{M}[n] = \begin{bmatrix} 0.00 & \mathbf{20.5} & 0.00 \\ \mathbf{16.3} & 0.00 & 0.00 \\ 0.00 & 0.00 & \mathbf{19.8} \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$P[n] = \begin{bmatrix} P_1[n] & P_2[n] & P_3[n] \end{bmatrix} \longrightarrow e^*(n) = \max_j P_j(n)$$

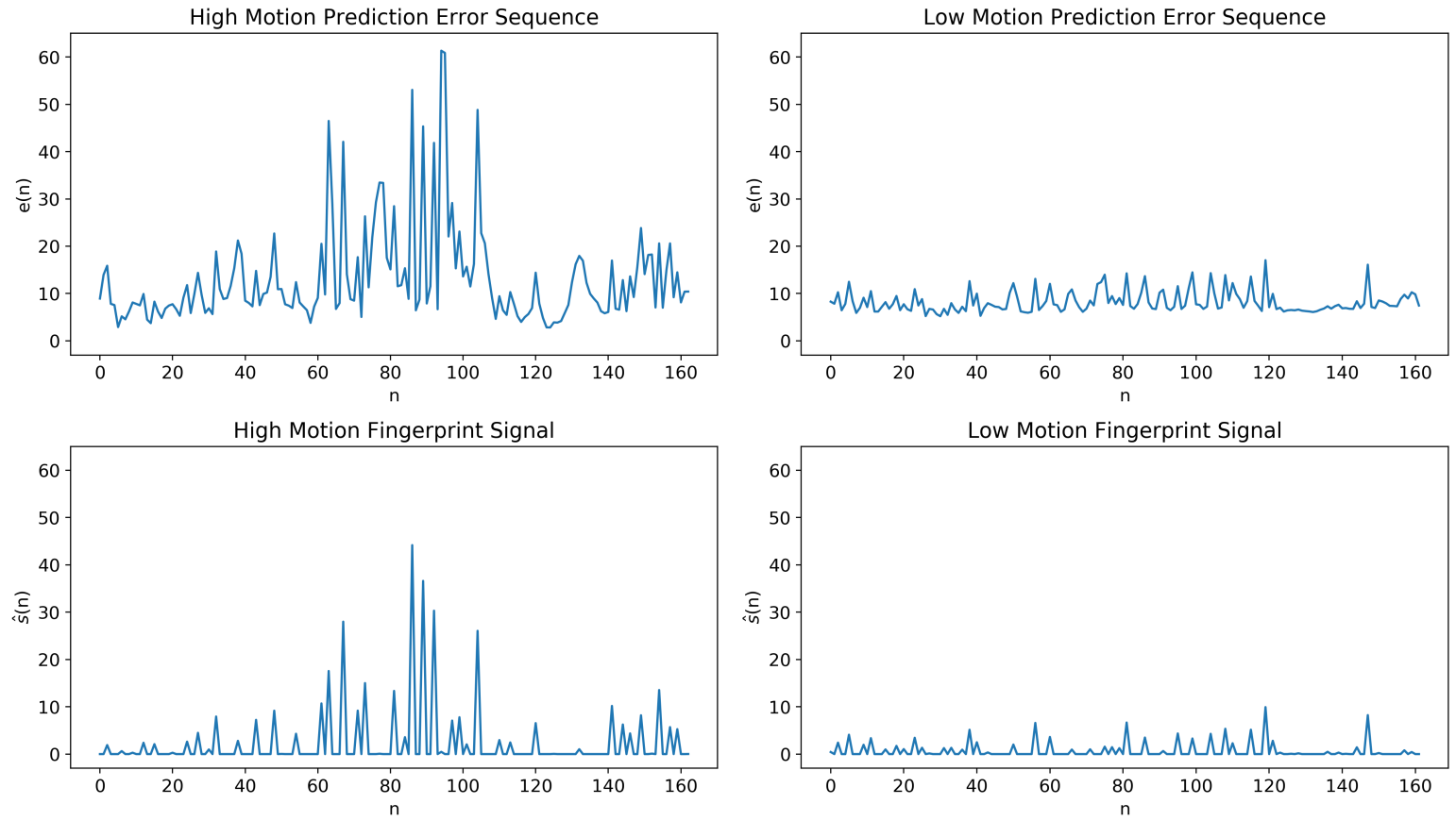
# Proposed Prediction Error Extraction Method



# New Detection Features



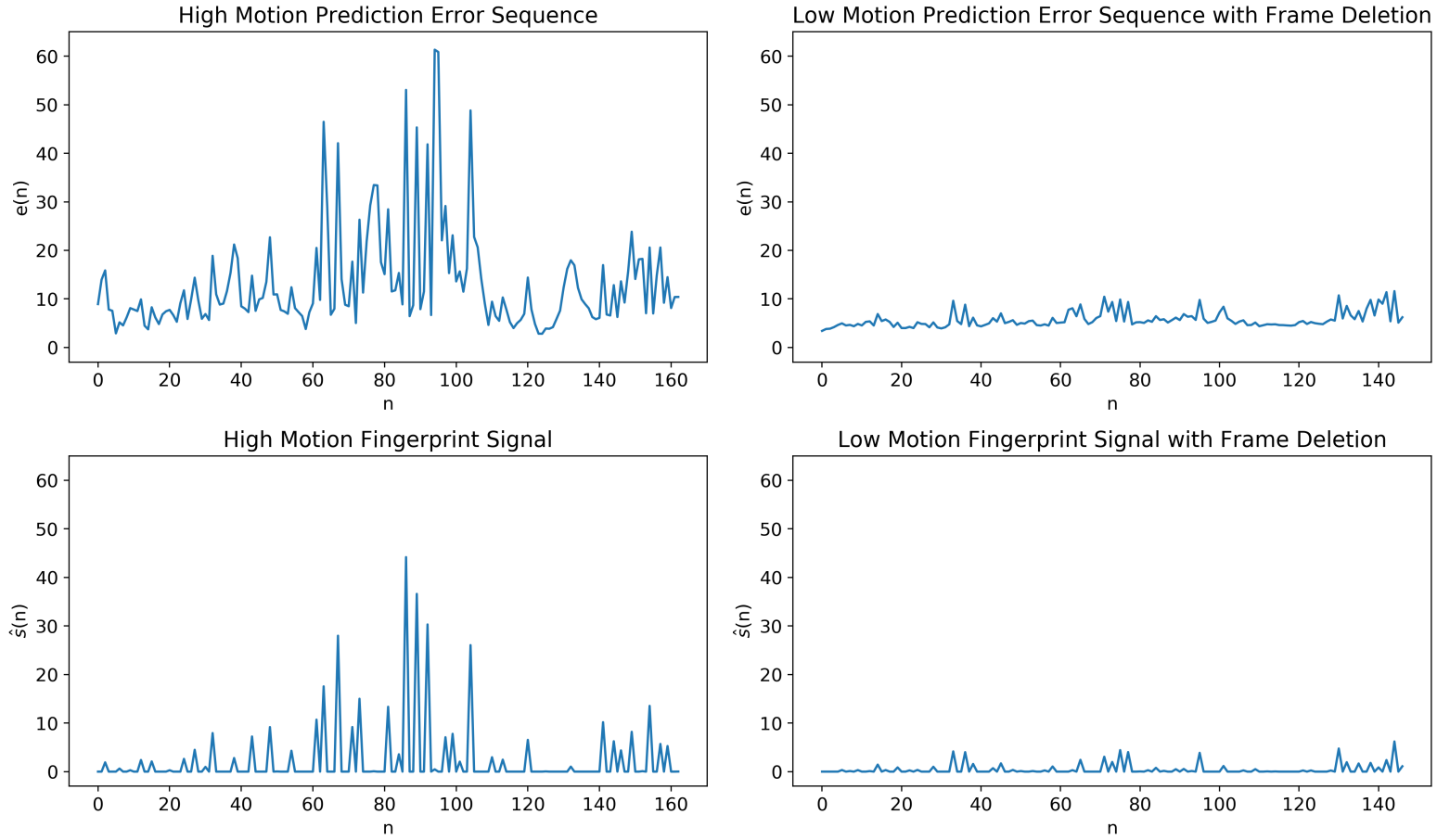
# New Detection Features




- Video capture conditions *affect* estimated fingerprint energy

# New Detection Features

- To further illustrate



# New Detection Features

- Differences in capture conditions can result in *misclassifications*
- Require some *statistical representation* of
  - The inferred prediction error sequence $e^*(n)$
  - The estimated fingerprint signal $\hat{s}(n)$
- Over the span of a few seconds, the sequences can be *wide-sense stationary*
- We propose modeling the sequences as an *autoregressive* (AR) process

# AR Models

- Often used for forecasting
- Predicts future values of the process using a *linear combination* of past observations and white noise

$$u(n) = \sum_{k=1}^{M} w_k^* u(n-k) + v$$

- The *variance* of the noise term indicates the quality of the fit

# New Detection Features

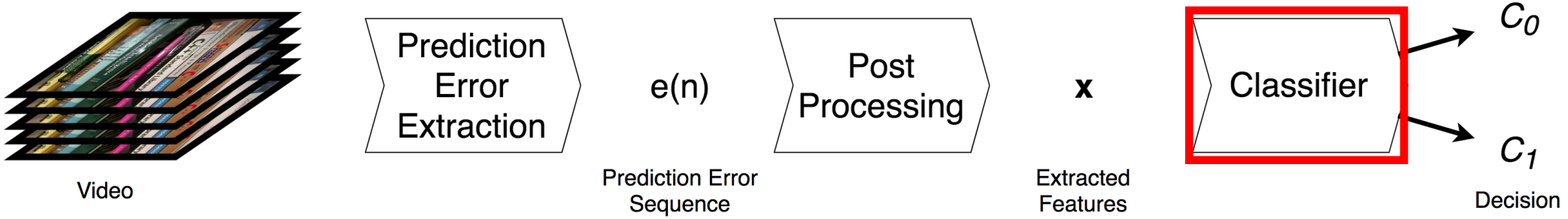
- Construct AR model of *both* sequences
  - Add AR model parameters
  - Add Noise variance
- In addition to the AR model parameters we add
  - The sample *mean* and *variance* of $e^*(n)$
  - The sample mean and variance of $\hat{s}(n)$
- These help *scale* the decision boundary

# New Detection Features

- Our final feature vector **x** is defined as

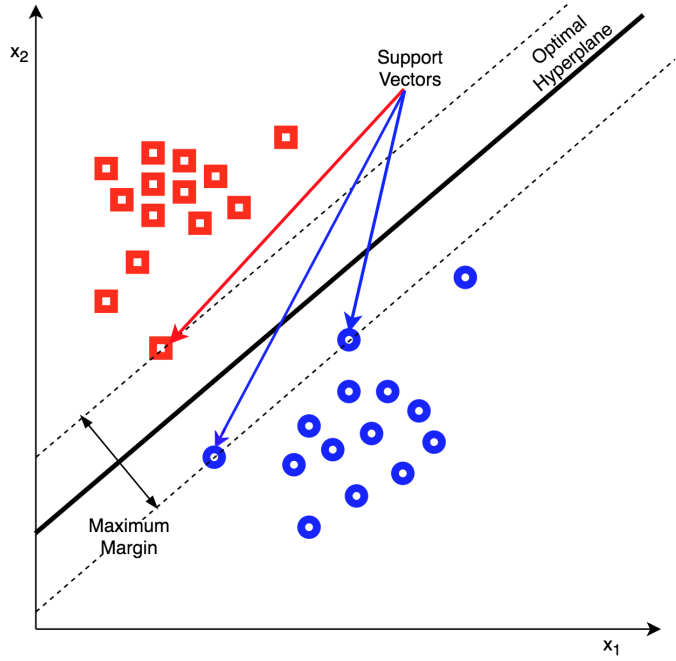
$$\boldsymbol{x} = \begin{bmatrix} \frac{1}{N}\sum_{n=1}^{N} |\hat{s}[n]| \\ \mu_{\hat{s}} \\ \sigma^2_{\hat{s}} \\ \mu_{e^*} \\ \sigma^2_{e^*} \\ \boldsymbol{w}_{\hat{s}} \\ \boldsymbol{w}_{e^*} \\ \sigma^2_{v(\hat{s})} \\ \sigma^2_{v(e^*)} \end{bmatrix}$$

# Proposed Classification Method



# Proposed Classification Method

- It is *difficult* to build parametric models of each feature
- Want to make use of all features in aggregate
- We propose using a *Support Vector Machine* (SVM) classifier



# Experimental Results

- We conducted a *series of experiments* to evaluate
  - The performance of our proposed prediction error extraction method
  - The performance of our enhanced feature set
- We constructed a *labeled dataset*
- The dataset was captured using seven camera models
  - Over 250 videos per camera model
  - ~5 seconds long
  - Variety of lighting conditions (indoor & outdoor)
  - Variety of motion and scene content
  - Variety of GOP structures

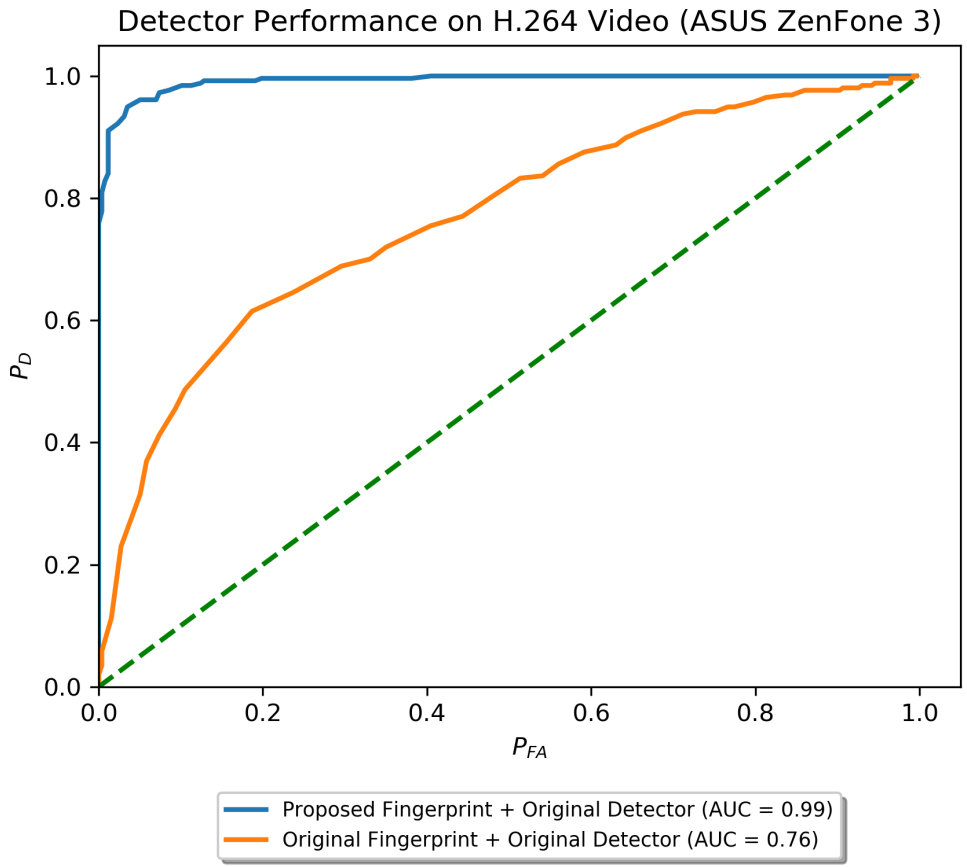
# Dataset

| Camera Model | GOP Type | GOP N | GOP M | Genuine Videos | Frame Deleted Videos |
|---|---|---|---|---|---|
| ASUS ZenFone 3 Laser | Fixed | 30 | 1 | 257 | 257 |
| Apple iPhone 8 plus | Variable | 30 | 2 | 248 | 245 |
| Google Pixel 1 | Variable | 29 | 2 | 776 | 777 |
| Google Pixel 2 | Variable | 29 | 2 | 212 | 212 |
| Kodak Ektra | Fixed | 30 | 1 | 503 | 502 |
| Nokia 6.1 | Fixed | 30 | 1 | 501 | 500 |
| Samsung Galaxy S7 | Fixed | 30 | 1 | 231 | 231 |
| **Total Count** | | | | **2728** | **2724** |

- Total of *5452* videos
- Google Pixel 1 video captured from 3 devices
- Kodak Ektra and Nokia 6.1 video captured from 2 devices

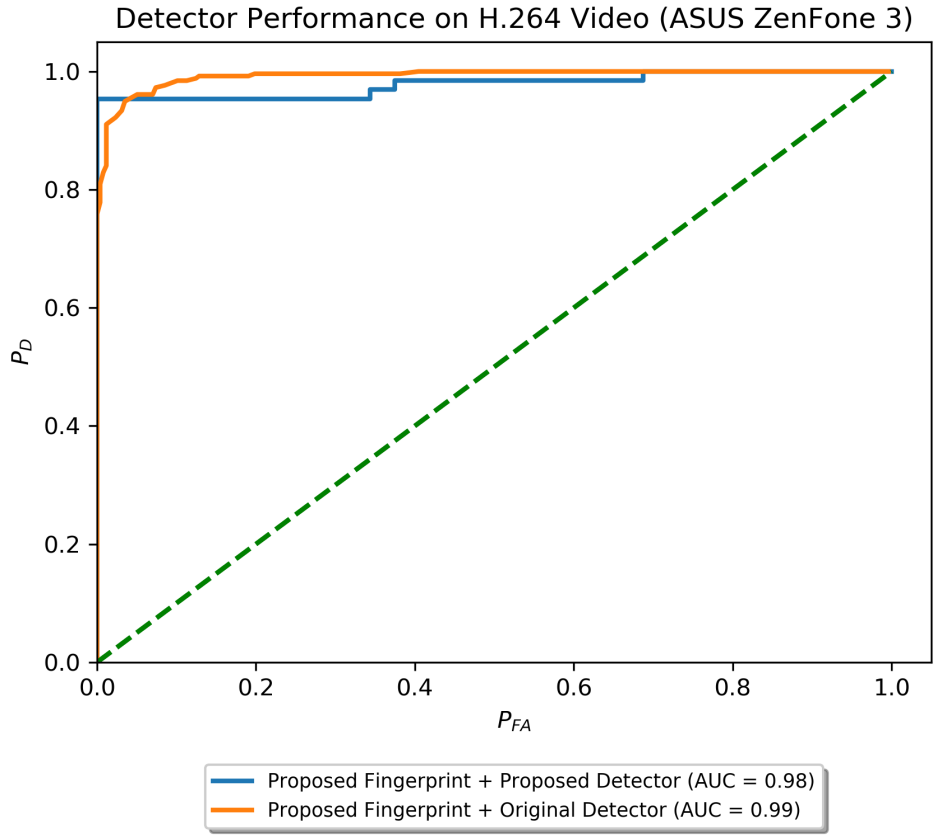
# Experimental Results (One Camera)

- Performance benchmarked using videos from *one camera model* (ASUS ZenFone 3 Laser)
- Energy based detector
- Proposed Extraction Method
  - AUC of 0.99 (v.s. 0.76)
  - $P_D$ of 0.9 at 5% $P_{FA}$ (v.s. 0.4)



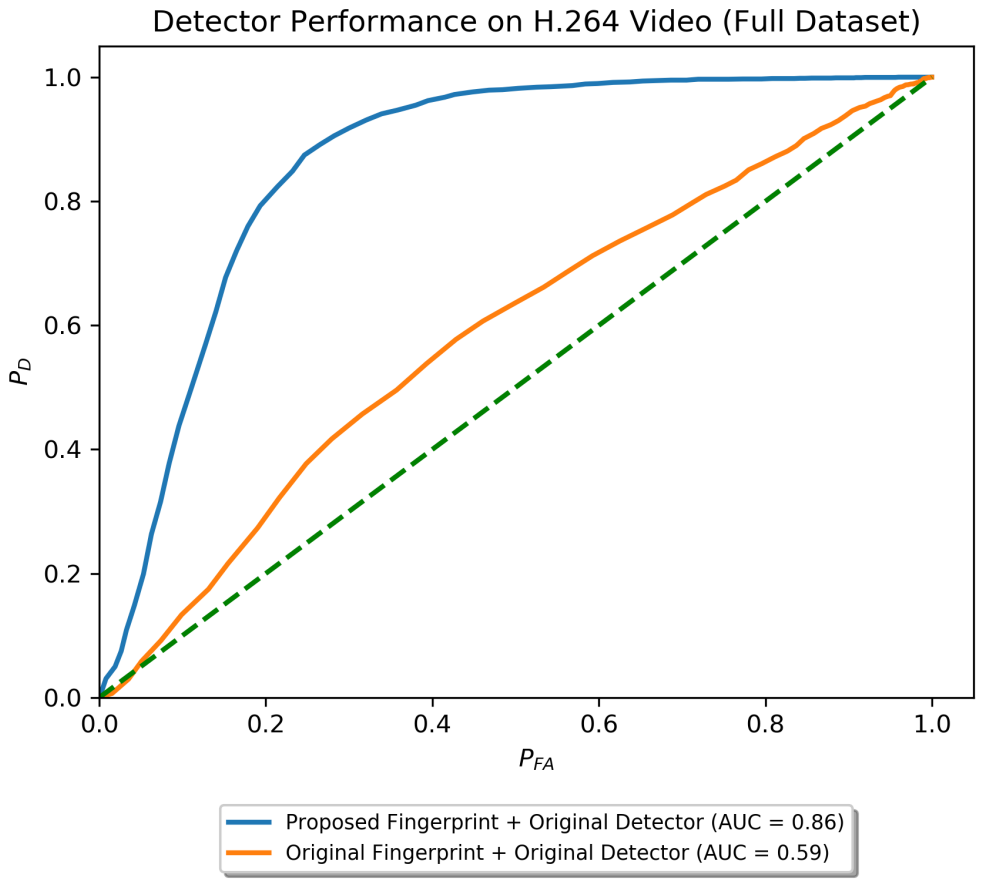
# Experimental Results (One Camera)

- SVM trained using videos from *one camera model*
  - 75-25 Train-test split
  - Radial Basis Kernel
  - AR Model order of 31
- Effects of Proposed Detector
  - $P_D$ of 0.95 at 1% $P_{FA}$ (v.s. 0.8)



# Experimental Results (All Cameras)

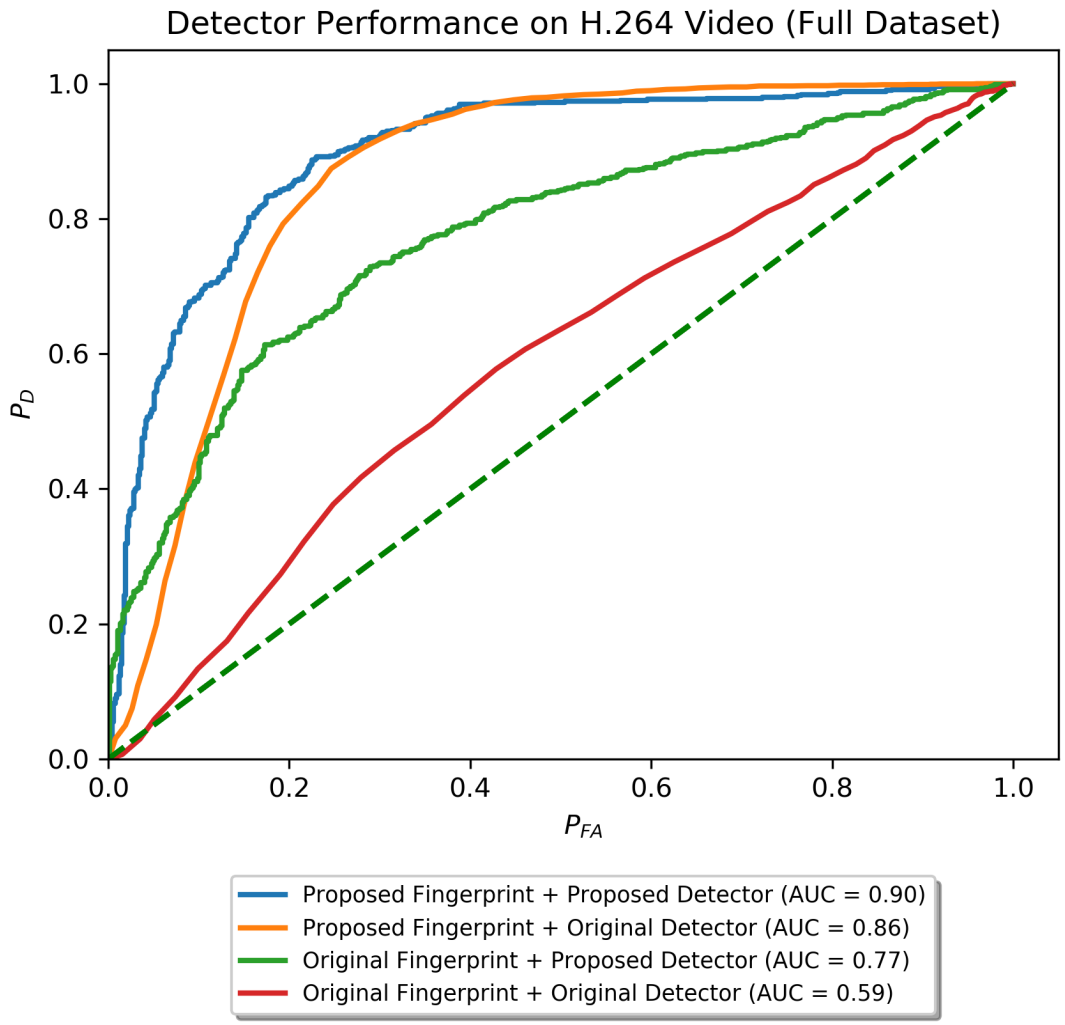
- Performance of detector evaluated using videos from *all camera models*
- Energy Based detector
- Proposed Extraction Method
  - AUC of 0.86 (v.s. 0.59)
  - $P_D$ of 0.5 at 10% $P_{FA}$ (v.s. 0.15)



# Experimental Results (All Cameras)

- Evaluate performance of proposed detector on all videos
- Extract prediction error using both *original* & *proposed* methods
- Generate feature vectors
- SVM trained using videos from *all camera models*
  - 75-25 Train-test split
  - Radial Basis Kernel
  - AR Model order of 31

# Experimental Results (All Cameras)



# Experimental Results (Summary)

- Original fingerprint + detector *fail* on diverse dataset
- Both proposed fingerprint & detector increase performance
- Proposed fingerprint + detector yield *best* performance
  - AUC of 0.9 (v.s. 0.59)
  - $P_D$ of *0.6* at 5% $P_{FA}$ (v.s. *0.05*)

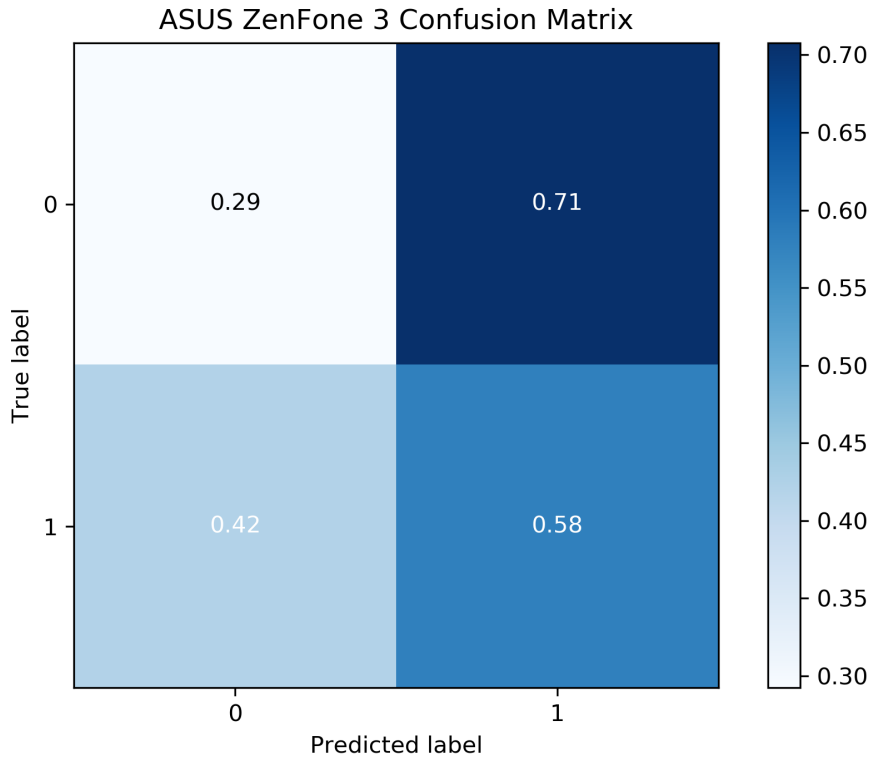
# Limitations

- Detector considered under *limited scenario*
- Plainly recompressed videos can exhibit *similar fingerprints*
- Given a feature vector **x** was found to have frame deletion

$$C_0 : \boldsymbol{x} \text{ resulted from an } \textit{altered} \text{ video.}$$
$$C_1 : \boldsymbol{x} \text{ resulted from a plainly } \textit{recompressed} \text{ video.}$$

# Limitations

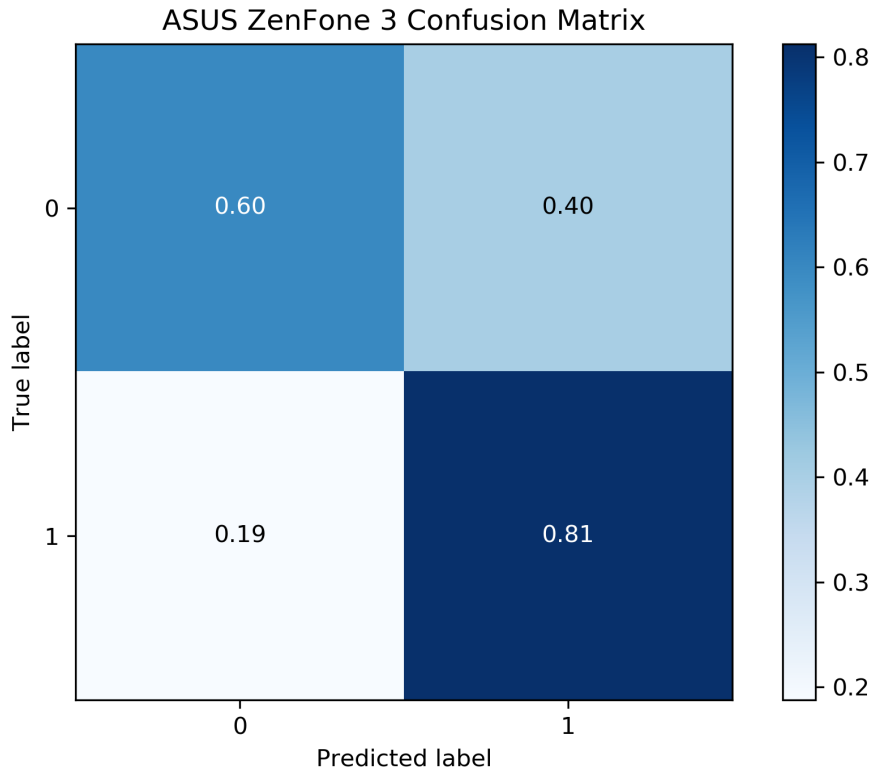
- Additional 257 plainly recompressed videos created
  - Using ASUS ZenFone 3
- Used Proposed fingerprint + detector
- Plotted *confusion matrix*




- Altered videos *confused* for recompressed videos

# Limitations

- Can alter system to improve accuracy
- Input observations scaled
  - Zero mean
  - Unit Variance




- Frame deletion features can provide some *differentiation*
- Incorporating additional solutions is *future work*

# Conclusion

- Original frame deletion fingerprints *not expressed* in H.264 video
- Proposed new methodology for H.264
  - New fingerprint extraction technique
  - New Classification Framework
- Evaluated new methods
  - ROC area *increased* by *0.31*
  - $P_D$ increased by *0.55* at 5% $P_{FA}$
- Noted system limitations, potential solutions, future work

# A New Approach to Detecting Frame Deletion in H.264 Encoded Digital Video

***Hunter Kippen***
Advisor: Dr. Matthew Stamm
Drexel University
hmk64@drexel.edu