Change-point Detection Methods for Body-Worn Video

Stephanie Allen, *SUNY Geneseo*
David Madras, *University of Toronto*
Ye Ye, *UCLA*
Greg Zanotti, *DePaul University*

Academic Mentor: Dr. Giang Tran
Consultant: Dr. Jeff Brantingham, *UCLA*
Industry Mentor: Sgt. Javier Macias, *LAPD*

August 18, 2016

LAPD & Body-Worn Video

- Third largest USA municipal police department, with 9,843 officers
- A leader in the effort to equip police officers with body-worn cameras

Body-worn Video (BWV)

Body-worn Video (BWV)

- Cameras worn on officers' chests used to record police-public interactions
  - Currently deployed to 1,200 officers; will be scaled up to 7,000
- **Benefits**:
  - Provide video record in the case of public disagreements
  - Shown to increase police professionalism
- **Challenge**:
  - Create large volumes of data, necessitating automatic data analysis

Problem Statement

- **Goal:** Create algorithms to detect change-points in body-worn video
  - This will greatly streamline the video review process

- For this project, we focus on a specific class of change-points:
  - **The moment at which an officer exits or enters their car**

Images from www.youtube.com

Data Analysis - In Car Examples

Images from www.youtube.com

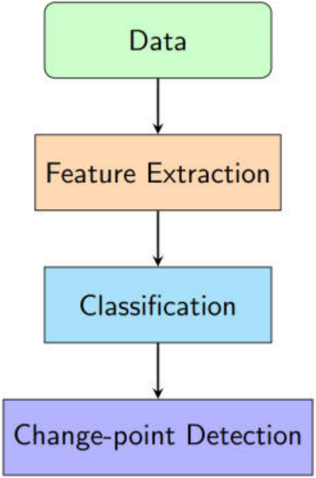
Data Analysis - Out of Car Examples

Images from www.youtube.com

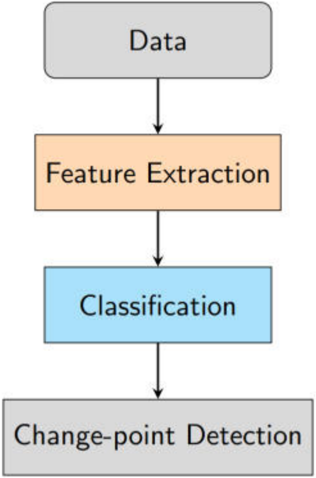
Data Analysis

- Sample of data taken from BWV pilot program (Dec '14-May '15)
- 691 videos, average length 9 minutes
- 420 contain either an entrance or exit from vehicle
- Of these:
  - 270 are taken from driver side
  - 274 are taken from a moving vehicle
  - 176 occur during nighttime

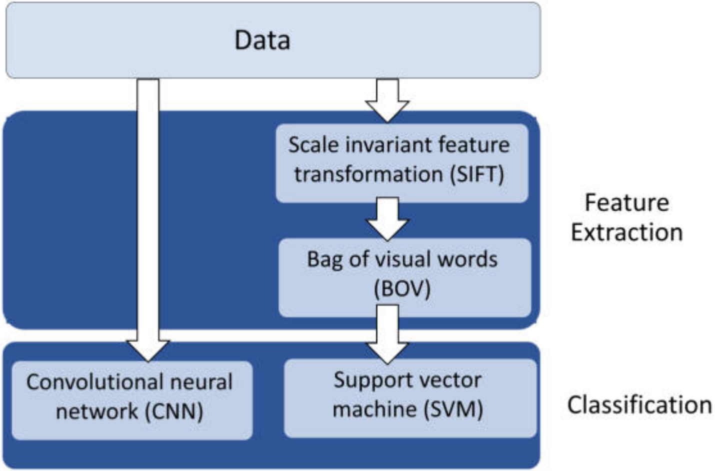
Overview of Methods



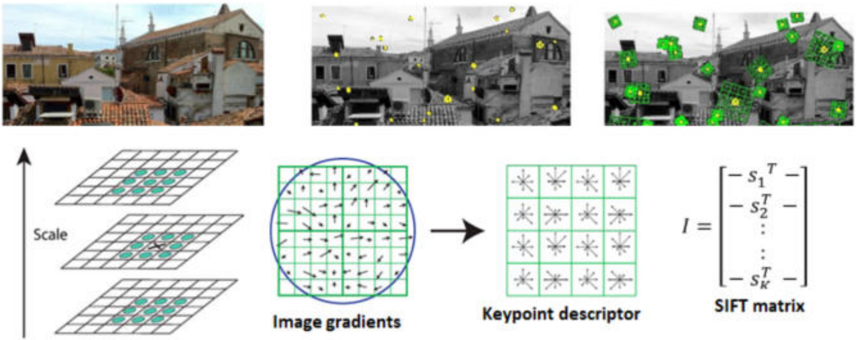
Overview of Methods - Feature Extraction & Classification



Overview of Methods - Feature Extraction & Classification



Keypoint Detection and Description – Scale-Invariant Feature Transformation (SIFT)



Images from Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", and VLFeat.org

Image Representation - Bag of Visual Words

- Sample 20% of images in the training set, extract SIFT descriptors
- Apply *k*-means clustering, where the centroid of each cluster is a 'visual word'

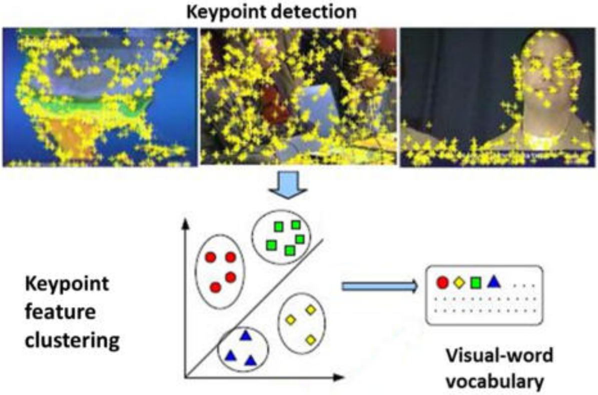
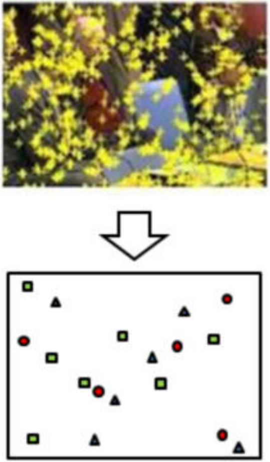


Image from Zhang et al., "Evaluating Bag-of-Visual-Words Representations in Scene Classification"

Bag of Visual Words and Spatial Pyramid

For each new input image

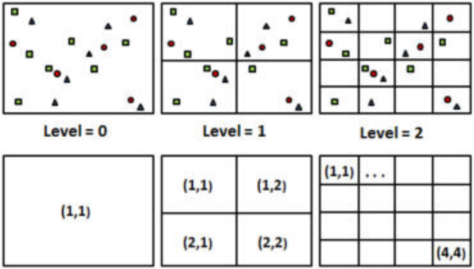
- Assign keypoint descriptors to nearest centroids



Bag of Visual Words and Spatial Pyramid

For each new input image

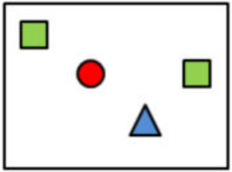
- Assign keypoint descriptors to nearest centroids
- Subdivide image into three levels of spatial resolution



Bag of Visual Words and Spatial Pyramid

For each new input image

- Assign keypoint descriptors to nearest centroids
- Subdivide image into three levels of spatial resolution
- Count # of descriptors for each spatial bin



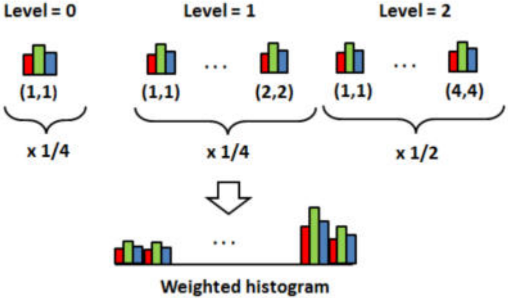
A spatial bin

Frequency histogram

Bag of Visual Words and Spatial Pyramid

For each new input image

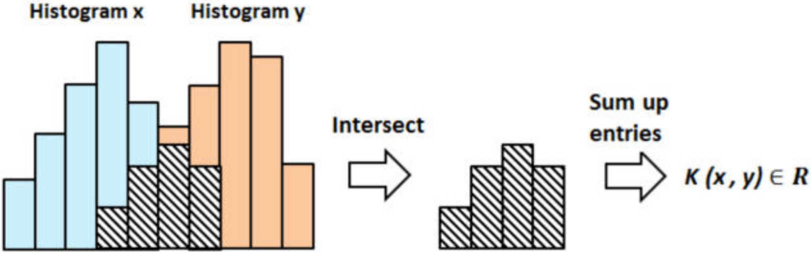
- Assign keypoint descriptors to nearest centroids
- Subdivide image into three levels of spatial resolution
- Count # of descriptors for each spatial bin
- Weight and concatenate spatial histograms



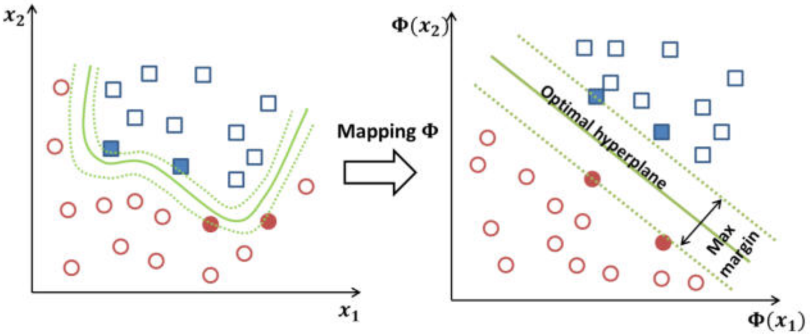
Histogram Intersection Kernel

- Goal: quantify similarity between two weighted histograms
- For two histograms $x, y \in \mathbb{R}^D$, kernel is defined as

$$K(x, y) = \sum_{i=1}^{D} \min(x_i, y_i).$$



Classifier - Support Vector Machine (SVM)



- Kernel function $K(x, y) = \Phi(x)^T \Phi(y) = \sum_{i=1}^{D} \min(x_i, y_i)$.
- Maximize margin and obtain weight coefficients
- For a new image histogram x, $Score(x) = \sum_{n=1}^{N} a_n t_n K(x, x_n) + b$

Classifier - Neural Network

- An artificial neural network jointly learns a **feature representation** and **discriminative classifier** over data
- Neurons are stacked on top of one another in **layers** to form complex, highly informative features
- At the last layer, outputs are normalized to form **class predictions**

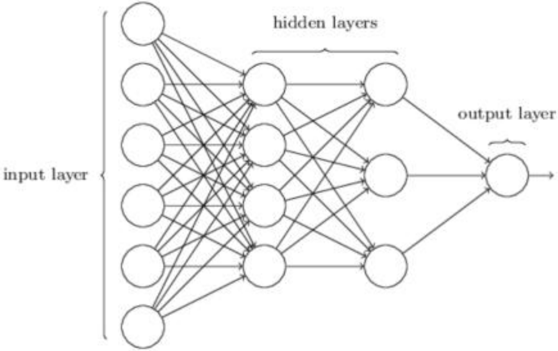
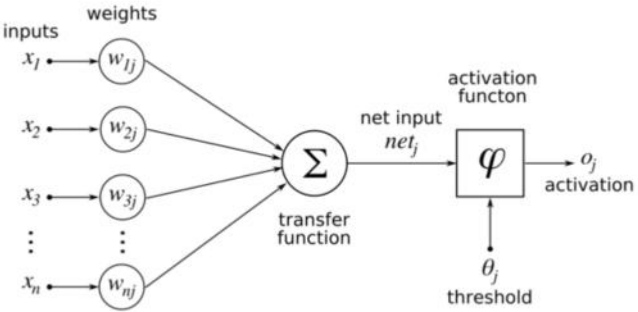


Image from Nielsen, *Neural Networks and Deep Learning*

Neural Network Detail

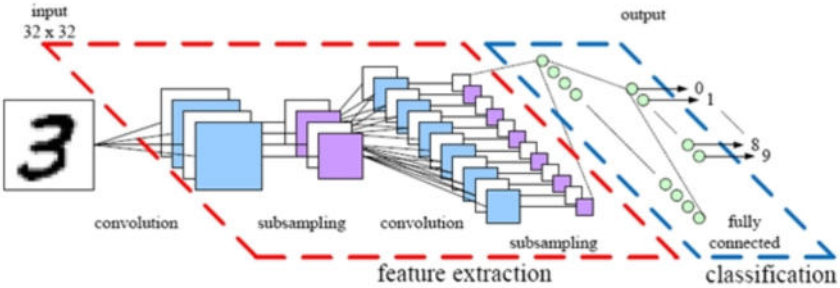
- Generally, operations within a neuron consist of **multiplying inputs by weights**, passing them to a **transfer function**, and passing the result through a **nonlinear, thresholded "activation" function**



- Neural networks are trained by changing the weights according to an iterative optimization algorithm like gradient descent

Image from `https://en.wikibooks.org/wiki/File:ArtificialNeuronModel_english.png`

Convolutional Neural Networks

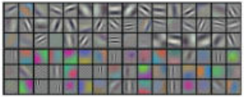


- Convolutional neural networks, or ConvNets, learn hierarchical filter banks for images. Architectures consist of alternating convolutional and pooling layers—some with nonlinearities.
- Convolutional layers slide a filter over an input to detect a certain pattern. Pooling layers subsample upstream outputs.

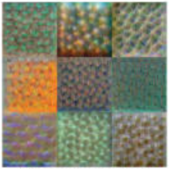
Image from Parallel Architecture Research Eindhoven

ConvNet Features

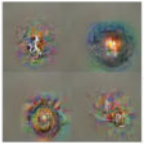
- As ConvNets are trained, the filters change what they detect and "learn" important features.
- Filters at early layers detect edges and blobs. Filters in later layers combine output of lower level filters to detect more complex patterns.



Conv 1: Edge+Blob



Conv 3: Texture



Conv 5: Object Parts

Fc8: Object Classes

Image from `http://www.cc.gatech.edu/~hays/compvision/proj6/`

Using and Finetuning ConvNets

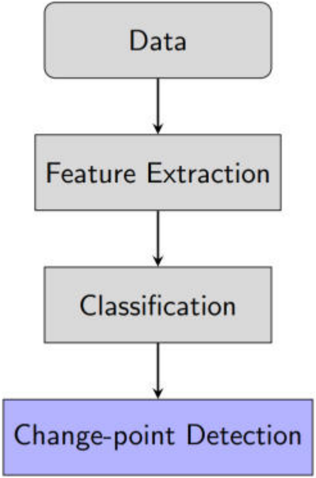
- Although ConvNets are extremely powerful, training them can be incredibly computationally intensive
- General convolutional networks for image recognition are created and released by researchers, and can be "finetuned" to specific problems
- We modify the popular VGG-16 architecture, and change only the top two layers to classify scenes as in/out of car

Classification Results

- Change-point detection depends on strong classification results
- Our predictions were made using 10-fold cross-validation on a large sample of or all of the videos
- Precision: How many of our out of car **predictions** were truly out of car?
- Recall: How many of our out of car **frames** did we correctly identify?

Classifier	**Accuracy**	**Precision**	**Recall**
SIFT-BOV-SVM	90%	92%	89%
ConvNet	**94%**	**96%**	**95%**

Overview of Methods - Change-point Detection



Change-point Methods Overview

- Given a time series $X_i, i = 1...n$, there may be one or more **change-points** c where the underlying distribution of the X_i changes.

- In the case of one change-point:
$$X_i \sim F_1 \; \forall \; i \leq c, \; X_i \sim F_2 \; \forall \; i \; > c$$
for some distributions $F_1 \neq F_2, c \in \{1...n\}$

- **Goal**: To find c
  - Evaluate an objective function or test statistic for each X_i for $i \in \{1...n\}$
  - Find i to optimize the objective function or all i which produce a test statistic value greater than a threshold

Five Change-point Methods

1. Forecasting/Time Series Analysis
2. BoVW Histogram Comparison
3. Hidden Markov Model
4. Mean-Squared Error
5. Maximum Likelihood

Method 1: Forecasting/Time Series Analysis

- Elements in a time series often are correlated with each other.

$$\text{Autoregressive One Lag (AR(1))} : X_t = B_0 + B_1 X_{t-1}$$

- Assume the sequence of scores is stationary between change-points — meaning the mean is constant during those intervals

- We can forecast the next observation in a given interval based on a mean of the previous observations.

$$\text{Mean Model} : X_t = \bar{X}$$

Method 1: Forecasting/Time Series

- "Future window" technique: Enables the application of forecasting methods to change-point detection
  - Estimate a model based on data-points from the beginning of the series
  - Forecast a set number of future values using the established model
  - If the forecasting error for **all of these observations** is larger than a set threshold, declare a change-point.
  - Re-estimate the model based on the observations in this window

Method 2: BoVW Histogram Comparison

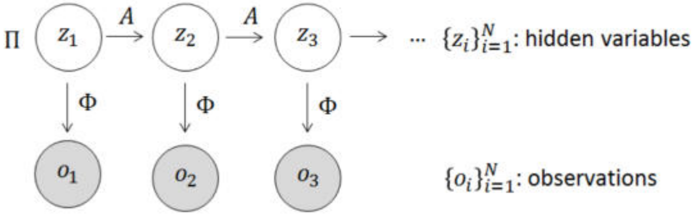
- Establish a baseline histogram and compare successive histograms in the series to this baseline via the future window technique:
  - χ^2 Method: $\chi^2 = \sum_{i=1}^{m} \frac{(o_i - e_i)^2}{e_i}$,
    where e is the baseline histogram and o is a histogram in the future window
  - Match Distance: $d_M(H, K) = \sum_{i=1}^{m} |h_i - k_i|$,
    where h_i is the cumulative histogram of the elements of h up to bin i, h is the baseline histogram, and k is a histogram in the future window

Method 3: Hidden Markov Model

- **Goal:** given a sequence of observations, infer the most probable sequence of hidden variables.
- **Change-point** = transitions in the inferred states of hidden variables

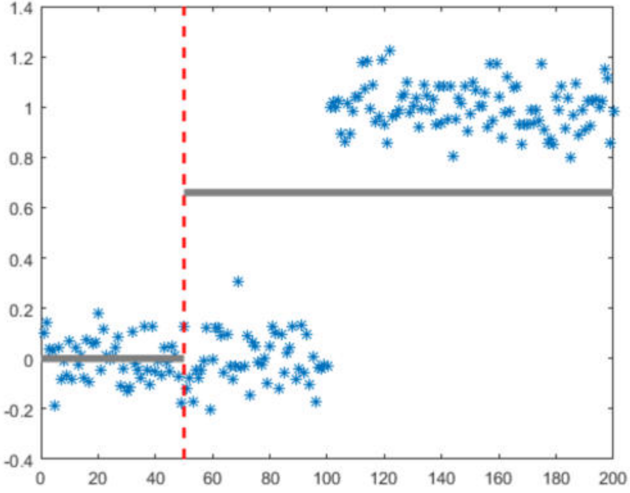
Method 3: Hidden Markov Model

- **Goal:** given a sequence of observations, infer the most probable sequence of hidden variables.
- **Change-point** = transitions in the inferred states of hidden variables

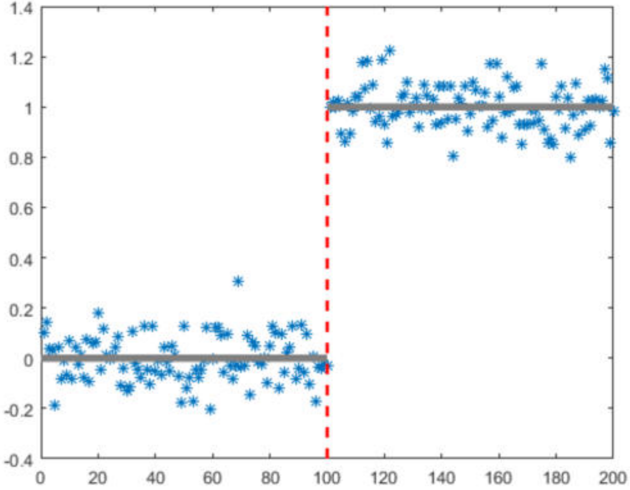


Π: initial distribution
A: transition matrix
Φ: emission parameters of observations' distributions

Method 4: Mean-Squared Error



Method 4: Mean-Squared Error



Method 4: Mean-Squared Error

- For large enough sample size, the sample mean $\bar{x}_i$ will be a **normal random variable** by the Central Limit Theorem
- Therefore, $\bar{x}_i^2$ will be a **gamma random variable** and:

$$MSE(c) - \sum_{i=1}^{n} x_i^2 = c\bar{x}_1^2 + (n-c)\bar{x}_2^2 \sim \boldsymbol{\Gamma(1, 2\sigma_x^2)}$$

- We can then derive a p-value for a measurement of mean-squared error

$$p = \frac{MSE(c) - \sum_{i=1}^{n} x_i^2}{2\sigma_x^2}$$

- Where p-value is low, we are near a change-point

Method 4: Mean-Squared Error

- We can now recursively extend mean-squared error to sequences with multiple change points
  1. Given sequence x_i, find x_j with smallest MSE.
  2. Calculate p-value for $MSE(j)$, then if $p \geq \alpha$ threshhold, stop.
  3. Run MSE again on sequences $x_1 ... x_{j-1}$ and $x_{j+1} ... x_n$.
  4. Return x_j, and the outputs of $MSE(x_1 ... x_{j-1})$ and $MSE(x_{j+1} ... x_n)$ as change-points.

Method 5: Maximum Likelihood Estimation

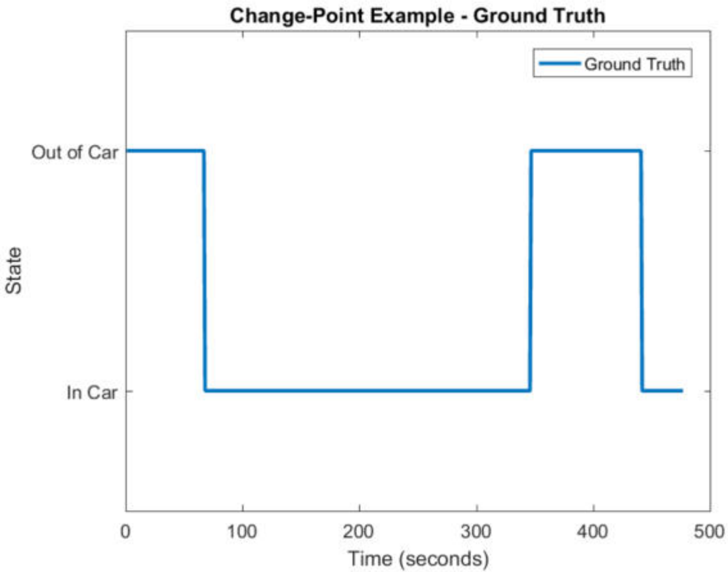
- We find the log-likelihood of the true labels given the data

$$\log \mathcal{L}(L, X) \sim \log \prod_{i=1}^{n} P(X_i | L_i)$$
$$= \log(p) \sum_{i=1}^{n} I[x_i = L_i] + \log(1-p) \sum_{i=1}^{n} I[x_i \neq L_i]$$

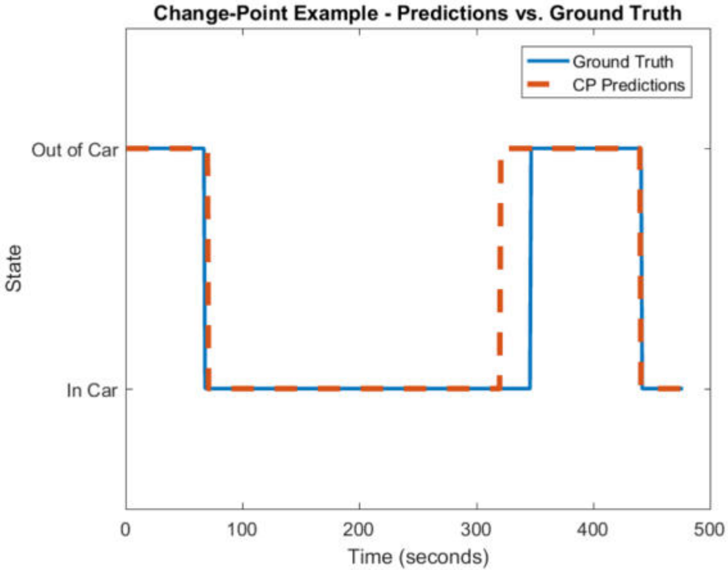
  where $x_i \in \{0, 1\}$ is classifier output, $p \in [0, 1]$ is classifier accuracy

- We maximize this likelihood by formulating it as a linear program, and constraining the number of possible change-points

Change-point Detection Results



Change-point Detection Results



Change-point Detection Results

- Using 691 LAPD videos (420 contain at least one change-point)
- Our methods ran on scores from the convolutional neural network

Table: Univariate Multiple Change-point Detection Results (All Videos)

Method	Recall (10 s)	Precision (10 s)
Autoregressive: One Lag	85%	60%
Maximum Likelihood	88%	61%
Mean Model	88%	61%
Mean-Squared Error	88%	**68%**
Hidden Markov Model	**93%**	65%

Change-point Detection Result - Multivariate Data

- Tested methods on BoVW histogram representations and CNN representations
- Representations were made in an **unsupervised** way—didn't need to train a classifier with labeled data (i.e. frames labeled in/out of car)
- **Benefits:** these methods are much more generalized
- **Challenges:** high-dimensional space is extremely complex, unsupervised methods are difficult to assess

Table: Multiple Change-point Detection Results for Multivariate Data

Method	Recall	Precision
Mean-Squared Error	86%	17%
Match Distance	98%	13%
χ^2 Test	**100%**	**20%**

Summary

- Annotated data, conducted data analysis
- Built and tuned classifiers to detect in car/out of car images with 90%+ accuracy, 95%+ precision and recall
- Developed a variety of change point detection methods for univariate and multivariate data
- Achieved 90% recall and nearly 70% precision on change-points in univariate data
- Methods work well on a variety of videos
  - With or without change-points
  - Driver or passenger side
  - Indoor or outdoor driving
  - Daytime or nighttime driving

Questions?

Future Work

- Improve unsupervised methods for multivariate time series
- Exploit the spatiotemporal structure of the data
- Explore applicability of change-point detection to other domains

Difference of Gaussians

- Subtract one blurred image from another less blurred image
- Increase visibility of edges

Original image

Image after difference of Gaussian filtering in black and white

Image from https://en.wikipedia.org/wiki/Difference_of_Gaussians

Histogram Intersection Kernel Proof

- Let $x, y \in \mathbb{R}^D$ be two histogram representations, and let M be the number of pixels in each image. Then, M is also an upper bound for the maximum number of keypoints in any image.
- Claim: A mapping function Φ can be found such that

$$\Phi(x)^T \Phi(y) = \sum_{i=1}^{D} \min(x_i, y_i).$$

- Proof by construction:

$$\Phi(x) := (\overbrace{1, 1, ..., 1,}^{x_1} \underbrace{0, 0, ..., 0,}_{M-x_1} \overbrace{1, 1, ..., 1,}^{x_2} \underbrace{0, 0, ..., 0,}_{M-x_2}$$

$$... \overbrace{1, 1, ..., 1,}^{x_D} \underbrace{0, 0, ..., 0}_{M-x_D})$$

VGG-16 Architecture

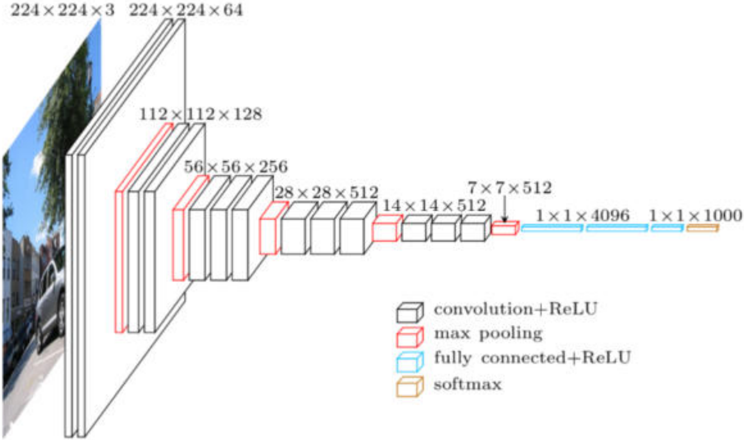


Image from
https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/

Hidden Markov Model

- Hidden variables $\{z_n\}_{n=1}^N$

$$z_n = \begin{cases} (1 \quad 0)^T & \text{if ``in-car''} \\ (0 \quad 1)^T & \text{otherwise} \end{cases}$$

- Initial distribution $\pi = (\pi_1 \quad \pi_2)$
- Transition probability $A_{ij} = p(z_{n,j} = 1 | z_{n-1,i} = 1)$, where $i, j \in \{1, 2\}$
- Conditional distributions of observed variables:

$$p(x_n | z_n, \Phi) = \left(\frac{1}{\sqrt{2\pi\sigma_1}} \exp(\frac{(x_n - \mu_1)^2}{\sigma_1}) \right)^{z_{n,1}} \cdot \left(\frac{1}{\sqrt{2\pi\sigma_2}} \exp(\frac{(x_n - \mu_2)^2}{\sigma_2}) \right)^{z_{n,2}},$$

where $\Phi = \{\sigma_1, \sigma_2, \mu_1, \mu_2\}$ is the set of emission parameters.

Hidden Markov Model Coefficient Estimates

- Initial distribution: $\hat{\pi} = [0.667 \quad 0.333]$
- Transition matrix: $\hat{A} = \begin{bmatrix} 0.9883 & 0.0117 \\ 0.0044 & 0.9956 \end{bmatrix}$
- Emission parameters:
  - Gaussian distribution governs the prediction of observed scores, based on the current state
  - In-car: $\hat{\mu_1} = -1.85$, $\hat{\sigma_1} = 1.33$
  - Out-of-car: $\hat{\mu_2} = 1.96$, $\hat{\sigma_2} = 1.06$

SIFT-BoVW-SVM Results

- The SVM scores were outputted for videos with change-points.

Table: Univariate Multiple Change-point Detection Results

Method	Recall (10 s)	Precision (10 s)
Maximum Likelihood Estimation	66%	**34%**
Autoregressive (1)	90%	17%
Hidden Markov Model	90%	17%
Mean Model	96%	18%
Mean-Squared Error	**91%**	30%

Change Point Detection Methods Applied to Body-Worn Video

Stephanie Allen, *SUNY Geneseo*
David Madras, *University of Toronto*
Ye Ye, *UCLA*
Greg Zanotti, *DePaul University*

Academic Mentor: Dr. Giang Tran
Consultant: Dr. Jeff Brantingham, UCLA
Industry Mentor: Sgt. Javier Macias, LAPD

August 18, 2016

LAPD & Body-Worn Video

- Third largest USA municipal police department, with 9,843 officers
- A leader in the effort to equip police officers with body-worn cameras

Body-worn Video (BWV)

Body-worn Video (BWV)

- Cameras worn on officers' chests used to record police-public interactions
  - Currently deployed to 1,200 officers; will be scaled up to 7,000
- **Benefits**:
  - Provide video record in the case of public disagreements
  - Shown to increase police professionalism
- **Challenge**:
  - Create large volumes of data, necessitating automatic data analysis

Problem Statement

- **Goal:** Create algorithms to detect change-points in body-worn video
  - This will greatly streamline the video review process

- For this project, we focus on a specific class of change-points:
  - **The moment at which an officer exits or enters their car**

Images from www.youtube.com

Data Analysis - In Car Examples

Images from www.youtube.com

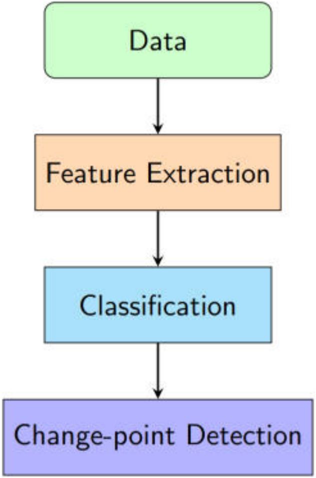
Data Analysis - Out of Car Examples

Images from www.youtube.com

Data Analysis

- Sample of data taken from BWV pilot program (Dec '14-May '15)
- 691 videos, average length 9 minutes
- 420 contain either an entrance or exit from vehicle
- Of these:
  - 270 are taken from driver side
  - 274 are taken from a moving vehicle
  - 176 occur during nighttime

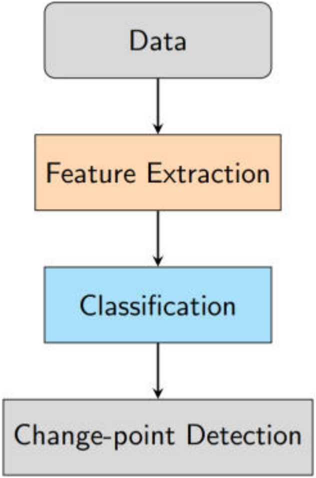
Overview of Methods



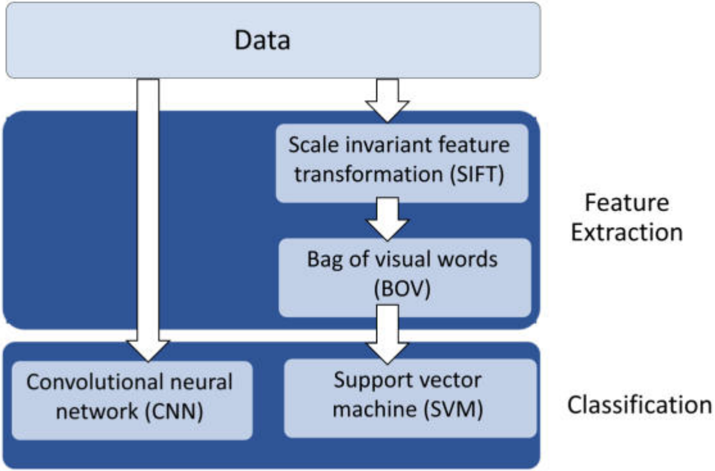
Overview of Methods

- **Feature extraction** methods take the sequence of images and reduce the images to compact representations that are then passed into **classifiers**.
- Other **classifiers** take raw images.
- **Change-point detection** methods have the ability to:
    - Take univariate or multivariate data
    - Detect any number of change-points per video

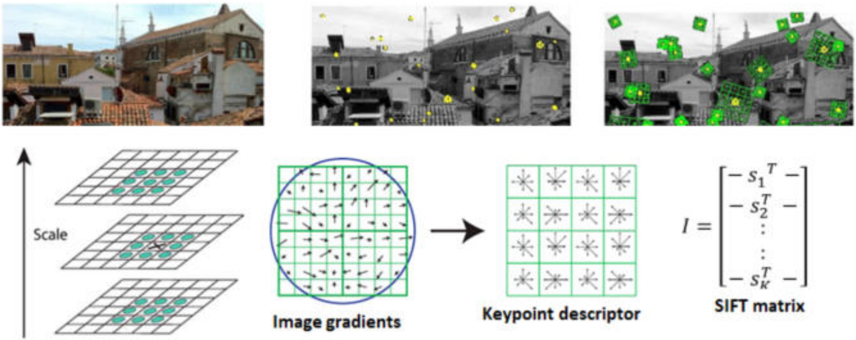
Overview of Methods - Feature Extraction & Classification



Overview of Methods - Feature Extraction & Classification



Keypoint Detection and Description – Scale-Invariant Feature Transformation (SIFT)



Images from Lowe, "Distinctive Image Features from Scale-Invariant Keypoints", and VLFeat.org

Image Representation - Bag of Visual Words

- Sample 20% of images in the training set, extract SIFT descriptors
- Apply k-means clustering, where the centroid of each cluster is a 'visual word'

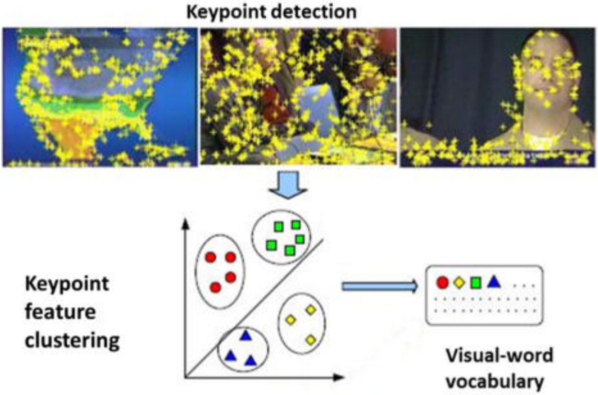
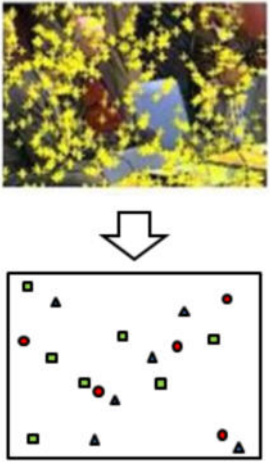


Image from Zhang et al., "Evaluating Bag-of-Visual-Words Representations in Scene Classification"

Bag of Visual Words and Spatial Pyramid

For each new input image

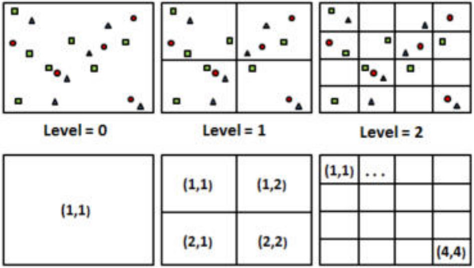
- Assign keypoint descriptors to nearest centroids



Bag of Visual Words and Spatial Pyramid

For each new input image

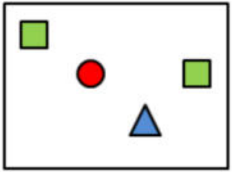
- Assign keypoint descriptors to nearest centroids
- Subdivide image into three levels of spatial resolution



Bag of Visual Words and Spatial Pyramid

For each new input image

- Assign keypoint descriptors to nearest centroids
- Subdivide image into three levels of spatial resolution
- Count # of descriptors for each spatial bin



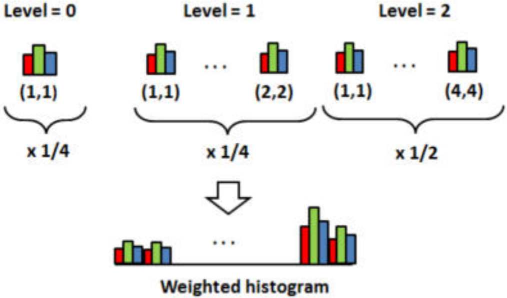
A spatial bin

Frequency histogram

Bag of Visual Words and Spatial Pyramid

For each new input image

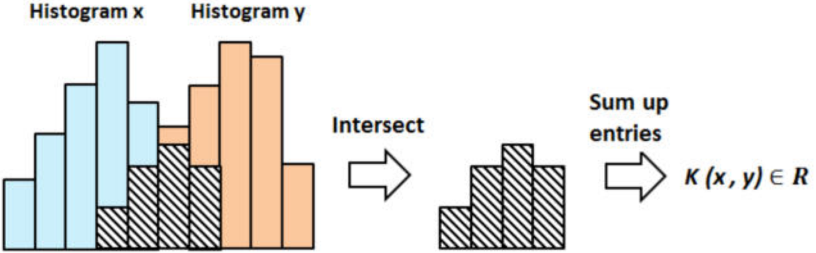
- Assign keypoint descriptors to nearest centroids
- Subdivide image into three levels of spatial resolution
- Count # of descriptors for each spatial bin
- Weight and concatenate spatial histograms



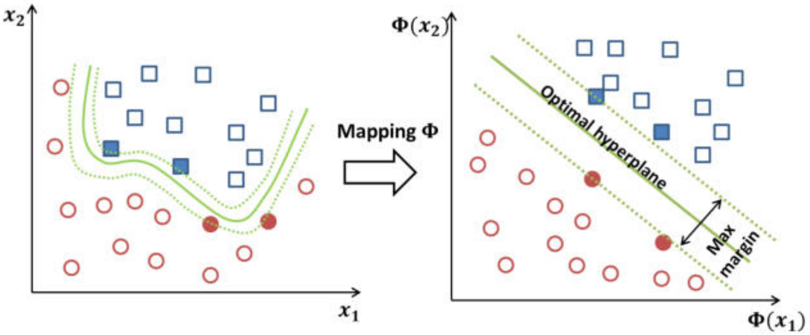
Histogram Intersection Kernel

- Goal: quantify similarity between two weighted histograms
- For two histograms $x, y \in \mathbb{R}^D$, kernel is defined as

$$K(x, y) = \sum_{i=1}^{D} \min(x_i, y_i).$$



Classifier - Support Vector Machine (SVM)



- Kernel function $K(x, y) = \Phi(x)^T \Phi(y) = \sum_{i=1}^{D} \min(x_i, y_i)$.
- Maximize margin and obtain weight coefficients
- For a new image histogram x, $Score(x) = \sum_{n=1}^{N} a_n t_n K(x, x_n) + b$

Classifier - Neural Network

- An artificial neural network jointly learns a **feature representation** and **discriminative classifier** over data
- Neurons are stacked on top of one another in **layers** to form complex, highly informative features
- At the last layer, outputs are normalized to form **class predictions**

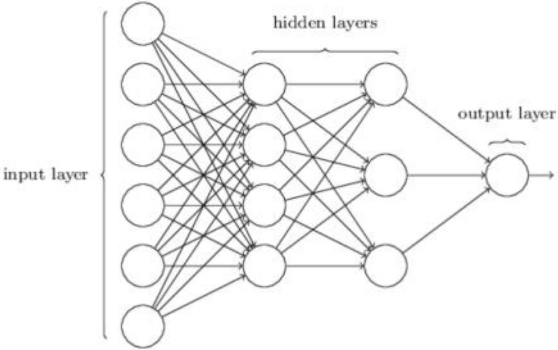
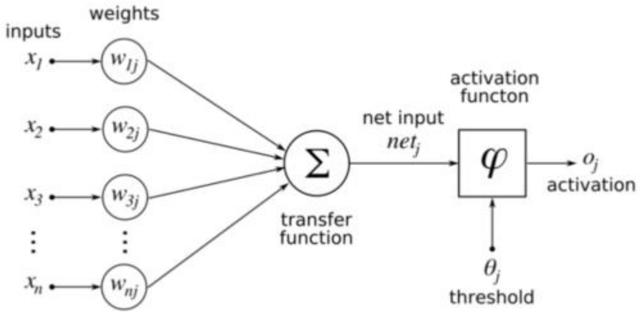


Image from Nielsen, *Neural Networks and Deep Learning*

Neural Network Detail

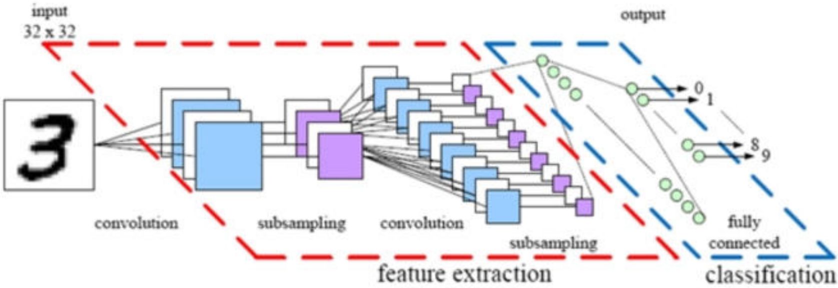
- Generally, operations within a neuron consist of **multiplying inputs by weights**, passing them to a **transfer function**, and passing the result through a **nonlinear, thresholded "activation" function**



- Neural networks are trained by changing the weights according to an iterative optimization algorithm like gradient descent

Image from `https://en.wikibooks.org/wiki/File:ArtificialNeuronModel_english.png`

Convolutional Neural Networks

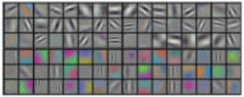


- Convolutional neural networks, or ConvNets, learn hierarchical filter banks for images. Architectures consist of alternating convolutional and pooling layers—some with nonlinearities.
- Convolutional layers slide a filter over an input to detect a certain pattern. Pooling layers subsample upstream outputs.

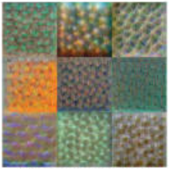
Image from Parallel Architecture Research Eindhoven

ConvNet Features

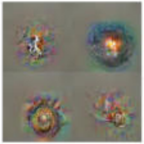
- As ConvNets are trained, the filters change what they detect and "learn" important features.
- Filters at early layers detect edges and blobs. Filters in later layers combine output of lower level filters to detect more complex patterns.



Conv 1: Edge+Blob



Conv 3: Texture



Conv 5: Object Parts

Fc8: Object Classes

Image from `http://www.cc.gatech.edu/~hays/compvision/proj6/`

Using and Finetuning ConvNets

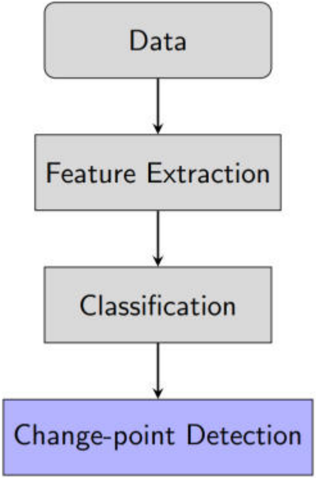
- Although ConvNets are extremely powerful, training them can be incredibly computationally intensive
- General convolutional networks for image recognition are created and released by researchers, and can be "finetuned" to specific problems
- We modify the popular VGG-16 architecture, and change only the top two layers to classify scenes as in/out of car

Classification Results

- Change-point detection depends on strong classification results
- Our predictions were made using 10-fold cross-validation on a large sample of or all of the videos
- Precision: How many of our out of car **predictions** were truly out of car? (complement of false pos. rate)
- Recall: How many of our out of car **frames** did we correctly identify?

Classifier	**Accuracy**	**Precision**	**Recall**
SIFT-BOV-SVM	90%	92%	89%
ConvNet	**94%**	**96%**	**95%**

Overview of Methods - Change-point Detection



Change-point Methods Overview

- Given a time series $X_i, i = 1...n$, there may be one or more **change-points** c where the underlying distribution of the X_i changes.
- In the case of one change-point:

$$X_i \sim F_1 \;\forall\; i \leq c,\; X_i \sim F_2 \;\forall\; i \;\; > c$$

  for some distributions $F_1 \neq F_2, c \in \{1...n\}$
- **Goal**: To find c
  - Evaluate an objective function or test statistic for each X_i for $i \in \{1...n\}$
  - Find i to optimize the objective function or all i which produce a test statistic value greater than a threshold

Five Change-point Methods

1. Forecasting/Time Series
2. BoVW Histogram Comparison
3. Hidden Markov Model
4. Mean-Squared Error
5. Maximum Likelihood

Method 1: Forecasting/Time Series

- Elements in a time series often are correlated with each other.

$$\text{Autoregressive One Lag (AR(1))} : X_t = B_0 + B_1 X_{t-1}$$

- If there are no change-points in a sequence of scores, we can assume the sequence is stationary and thus has a constant mean.

- We can forecast the next observation based on a mean of the previous observations.

$$\text{Mean Model} : X_t = \bar{X}$$

Method 1: Forecasting/Time Series

- "Future window" technique: Enables the application of forecasting methods to change-point detection
  - Estimate a model based on data-points from the beginning of the series
  - Forecast a set number of future values using the established model
  - If the forecasting error for **all of these observations** is larger than a set threshold, declare a change-point.
  - Re-estimate the model based on the observations in this window

Method 2: BoVW Histogram Comparison

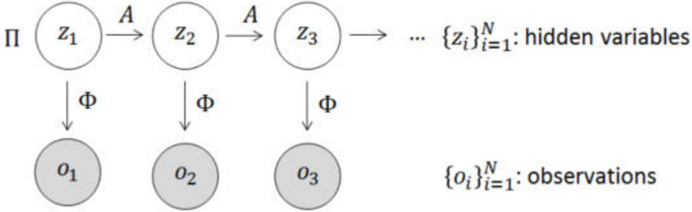
- Establish a baseline histogram and compare successive histograms in the series to this baseline via the future window technique:
  - χ^2 Method: $\chi^2 = \sum_{i=1}^{k} \frac{(o_i - e_i)^2}{e_i}$,
    where e is the baseline histogram and o is a histogram in the future window
  - Match Distance: $d_M(H, K) = \sum_{i=1}^{k} |h_i - k_i|$,
    where h_i is the cumulative histogram of the elements of h up to bin i

Method 3: Hidden Markov Model

- **Goal:** given a sequence of observations, infer the most probable sequence of hidden variables.
- **Change-point** = transitions in the inferred states of hidden variables

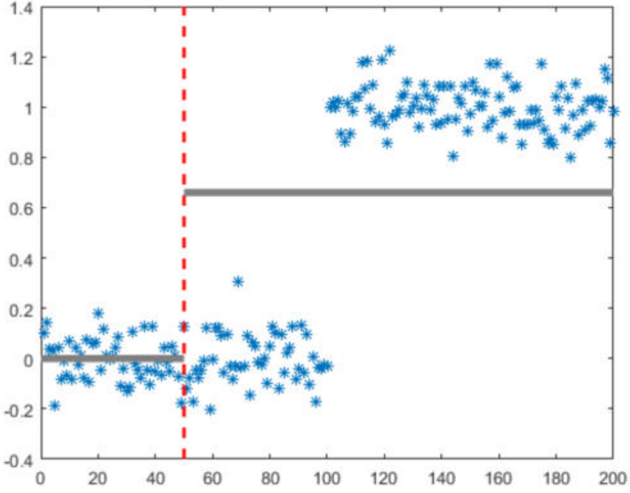
Method 3: Hidden Markov Model

- **Goal:** given a sequence of observations, infer the most probable sequence of hidden variables.
- **Change-point** = transitions in the inferred states of hidden variables

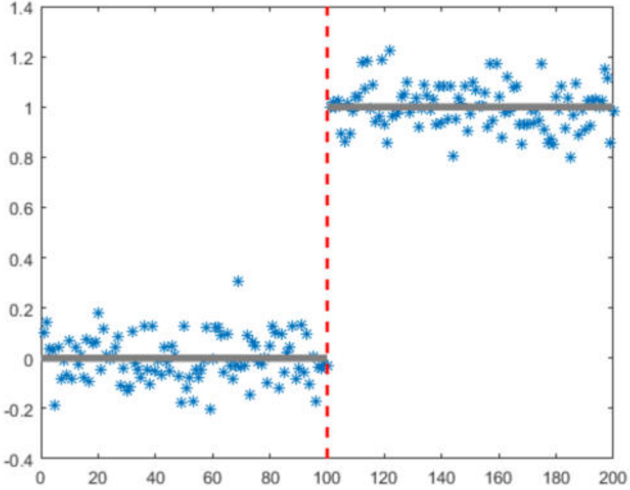


Π: initial distribution
A: transition matrix
Φ: emission parameters of observations' distributions

Method 4: Mean-Squared Error Change-point Detection



Method 4: Mean-Squared Error Change-point Detection



Method 4: Mean-Squared Error Change-point Detection

- For large enough samples, the sample mean $\bar{x}_i$ will be a **normal random variable** by the Central Limit Theorem
- Therefore, $\bar{x}_i^2$ will be a **gamma random variable** and:

$$MSE(c) - \sum_{i=1}^{n} x_i^2 = c\bar{x}_1^2 + (n-c)\bar{x}_2^2 \sim \Gamma(1, 2\sigma_x^2)$$

- We can then derive a p-value for a measurement of mean-squared error

$$p = \frac{MSE(c) - \sum_{i=1}^{n} x_i^2}{2\sigma_x^2}$$

- Where p-value is low, we are near a change-point

Method 4: Mean-Squared Error - Multiple Change-point Detection

- We can now recursively extend mean-squared error to sequences with multiple change points
  1. Given sequence x_i, find x_j with smallest MSE.
  2. Calculate p-value for $MSE(j)$, then if $p \geq \alpha$ threshhold, stop.
  3. Run MSE again on sequences $x_1 ... x_{j-1}$ and $x_{j+1} ... x_n$.
  4. Return x_j, and the outputs of $MSE(x_1 ... x_{j-1})$ and $MSE(x_{j+1} ... x_n)$ as change-points.

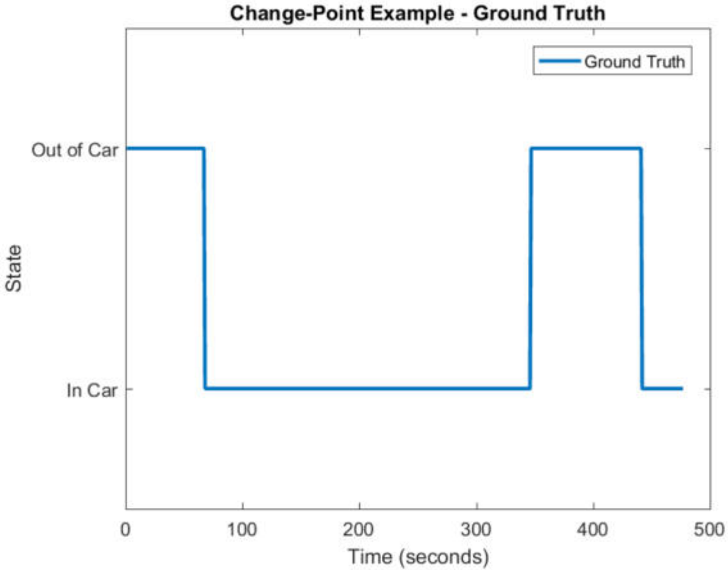
Method 5: Maximum Likelihood Estimation

- We find the log-likelihood of the true labels given the data

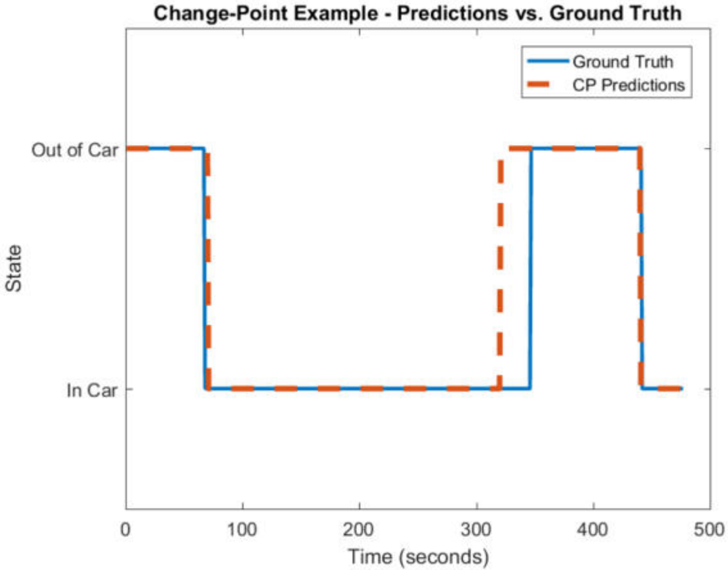
$$\log \mathcal{L}(L, X) \sim \log \prod_{i=1}^{n} P(X_i | L_i)$$
$$= \log(p) \sum_{i=1}^{n} I[x_i = L_i] + \log(1-p) \sum_{i=1}^{n} I[x_i \neq L_i]$$

  where $x_i \in \{0, 1\}$ is classifier output, $p \in [0, 1]$ is classifier accuracy
- We maximize this likelihood by formulating it as a linear program, and constraining the number of possible change-points

Change-point Detection Results



Change-point Detection Results



Change-point Detection Results

- Using 691 LAPD videos (420 contain at least 1 change-point)
- Our methods ran on output from the convolutional neural network

Table: Univariate Multiple Change-point Detection Results (All Videos)

Method	Recall (10 s)	Precision (10 s)
Autoregressive (1)	85%	60%
Maximum Likelihood	88%	61%
Mean Model	88%	61%
Mean-Squared Error	88%	**68%**
Hidden Markov Model	**93%**	65%

Change-point Detection Result - Multivariate Data

- Tested methods on BoVW histogram representations and CNN representations
- Representations were made in an **unsupervised** way—didn't need to train a classifier with labeled data (i.e. frames labeled in/out of car)
- Benefits: these methods are much more generalized
- Challenges: high-dimensional space is extremely complex, unsupervised methods are difficult to assess

Table: Multiple change-point detection Results for Multivariate Data

Method	**Recall**	**Precision**
Mean-Squared Error	86%	17%
Match Distance	99%	15%
χ^2 Test	**100%**	**21%**

Summary

- Annotated data, conducted data analysis
- Built and tuned classifiers to detect in car/out of car images with 90%+ accuracy, 95%+ precision and recall
- Developed a variety of change point detection methods for univariate and multivariate data
- Achieved 90% recall and nearly 70% precision on change-points in univariate data
- Methods work well on a variety of videos
  - With or without change-points
  - Driver or passenger side
  - Indoor or outdoor driving
  - Daytime or nighttime driving

Suggestions for RIPS 2017

- Improve unsupervised methods for multivariate time series
- Investigate methods for online data
- Exploit the spatiotemporal structure in the data
- Explore applicability of change-point detection to alternative domains

Questions?

Difference of Gaussians

- Subtract one blurred image from another less blurred image
- Increase visibility of edges

Original image

Image after difference of Gaussian filtering in black and white

Image from https://en.wikipedia.org/wiki/Difference_of_Gaussians

Histogram Intersection Kernel Proof

- Let $x, y \in \mathbb{R}^D$ be two histogram representations, and let M be the number of pixels in each image. Then, M is also an upper bound for the maximum number of keypoints in any image.
- Claim: A mapping function Φ can be found such that

$$\Phi(x)^T \Phi(y) = \sum_{i=1}^{D} \min(x_i, y_i).$$

- Proof by construction:

$$\Phi(x) := (\overbrace{1, 1, ..., 1,}^{x_1} \underbrace{0, 0, ..., 0,}_{M - x_1} \overbrace{1, 1, ..., 1,}^{x_2} \underbrace{0, 0, ..., 0,}_{M - x_2}$$

$$... \overbrace{1, 1, ..., 1,}^{x_D} \underbrace{0, 0, ..., 0}_{M - x_D})$$

VGG-16 Architecture

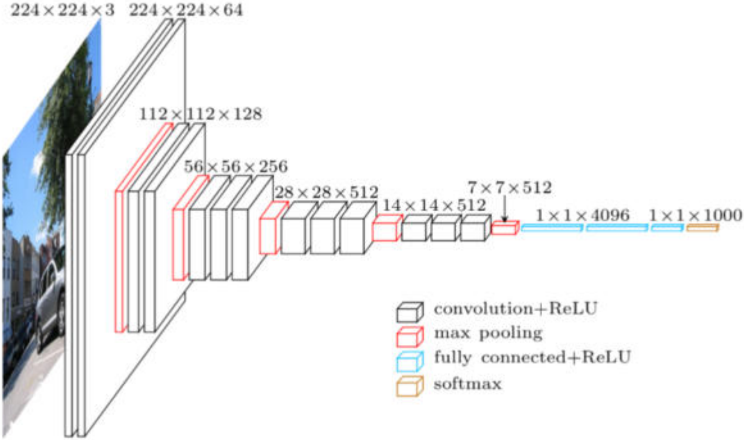


Image from
https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/

Hidden Markov Model

- Hidden variables $\{z_n\}_{n=1}^N$

$$z_n = \begin{cases} (1 \quad 0)^T & \text{if "in-car"} \\ (0 \quad 1)^T & \text{otherwise} \end{cases}$$

- Initial distribution $\pi = (\pi_1 \quad \pi_2)$
- Transition probability $A_{ij} = p(z_{n,j} = 1 | z_{n-1,i} = 1)$, where $i, j \in \{1, 2\}$
- Conditional distributions of observed variables:

$$p(x_n | z_n, \Phi) = \left(\frac{1}{\sqrt{2\pi\sigma_1}} \exp(\frac{(x_n - \mu_1)^2}{\sigma_1}) \right)^{z_{n,1}} \cdot \left(\frac{1}{\sqrt{2\pi\sigma_2}} \exp(\frac{(x_n - \mu_2)^2}{\sigma_2}) \right)^{z_{n,2}},$$

where $\Phi = \{\sigma_1, \sigma_2, \mu_1, \mu_2\}$ is the set of emission parameters.

Hidden Markov Model Coefficient Estimates

- Initial distribution: $\hat{\pi} = [0.667 \quad 0.333]$
- Transition matrix: $\hat{A} = \begin{bmatrix} 0.9883 & 0.0117 \\ 0.0044 & 0.9956 \end{bmatrix}$
- Emission parameters:
  - Standard deviations: $\hat{\sigma_1} = 1.3251$, $\hat{\sigma_2} = 1.0583$
  - Means: $\hat{\mu_1} = -1.8499$, $\hat{\mu_2} = 1.9646$

SIFT-BoVW-SVM Results

- The SVM scores were outputted for videos with change-points.

Table: Univariate Multiple Change-point Detection Results

Method	Recall (10 s)	Precision (10 s)
Maximum Likelihood	66%	34%
Mean Model	89%	18%
Autoregressive (1)	90%	17%
Hidden Markov Model	**90%**	17%
Mean-Squared Error	91%	**30%**