Linear Algebra in Forensics

Team Members:

Aditi Jain	PES2UG20CS021
Aishani Pandey	PES2UG20CS027
Amisha Mathew	PES2UG20CS038
Anirudh Chakravarty K	PES2UG20CS049

Problem statement

Understanding the concepts of linear algebra through real-world examples drawn from the field of forensics

Table of contents

Resources

- “Matching Handwriting with Its Author”
  Ziran Jiang, Aditya R. Mundada

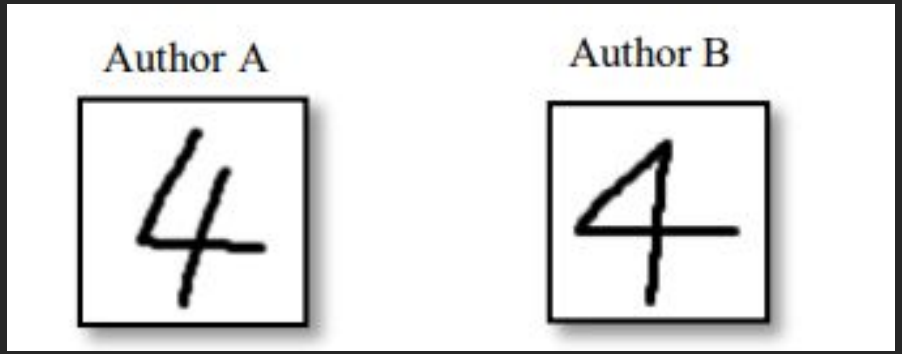
- Connecting Forensics and Linear Algebra
  By Donna Beers, Simmons College,
  Catherine Crawford, Elmhurst College

Handwriting Analysis

- Data Acquisition
- Image Processing using Naive Baye’s & SVM algorithm
- Algorithm comparison and Optimisation

01

Naive Baye's Algorithm



- Naive Bayes algorithm is a probabilistic classifier.
- It is a simple and effective technique used for quick predictions.
- It is based on Bayes theorem which is given as:

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)}$$

A, B	= events
$P(A\|B)$	= probability of A given B is true
$P(B\|A)$	= probability of B given A is true
$P(A), P(B)$	= the independent probabilities of A and B

- These 2 equations are used for making predictions on new sample.
- Here, y=1 means the author of the sample handwriting is A and y=0 means that the author is B.
- P(y=0)=P(y=1)=0.5

$$p(y = 1|x) = \frac{p(x|y=1)p(y=1)}{p(x)} \quad \dots \quad (2)$$

$$p(y = 0|x) = \frac{p(x|y=0)p(y=0)}{p(x)} \quad \dots \quad (3)$$

- n= total number of pixels in an image sample.
- Xi = ith pixel.

$$p(x|y=1) = \prod_{i=1}^{n} p(x_i|y=1) \quad \dots \quad (4)$$

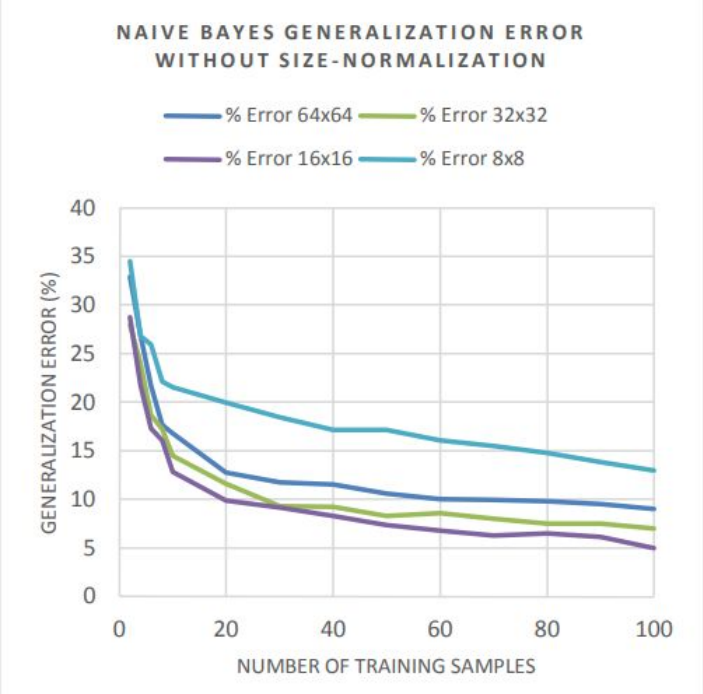
$$p(x|y=0) = \prod_{i=1}^{n} p(x_i|y=0) \quad \dots \quad (5)$$

- m= total number of training samples.
- Xj =Value of jth pixel in the ith training set.

$$p(x_j|y=1) = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)}=1 \wedge y^{(i)}=1\}+1}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}+2} \quad \dots \quad (6)$$

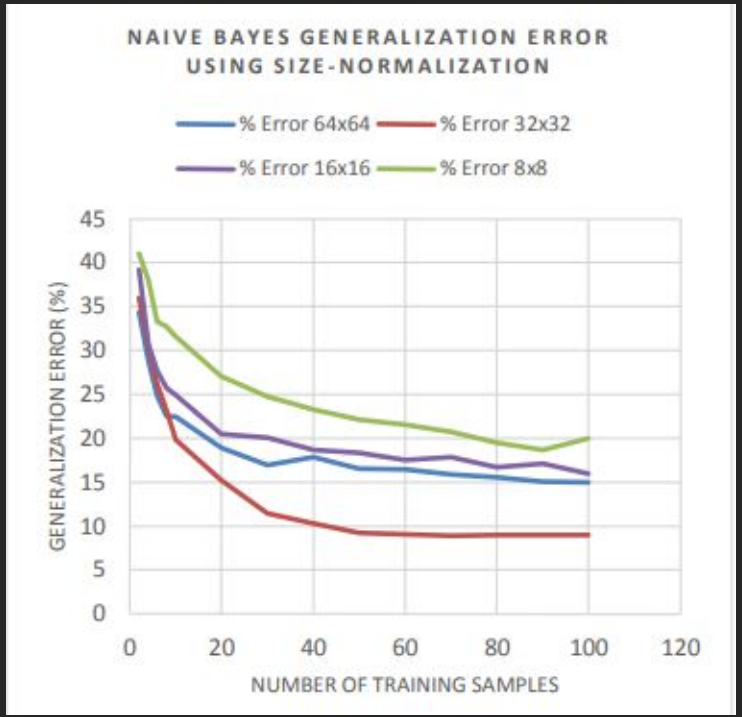
$$p(x_j|y=0) = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)}=1 \wedge y^{(i)}=0\}+1}{\sum_{i=1}^{m} 1\{y^{(i)}=0\}+2} \quad \dots \quad (7)$$

NB without size normalisation and centering



RESOLUTION	GENERALIZATION ERROR (%)
16x16	5
32x32	7
64x64	9
8x8	13

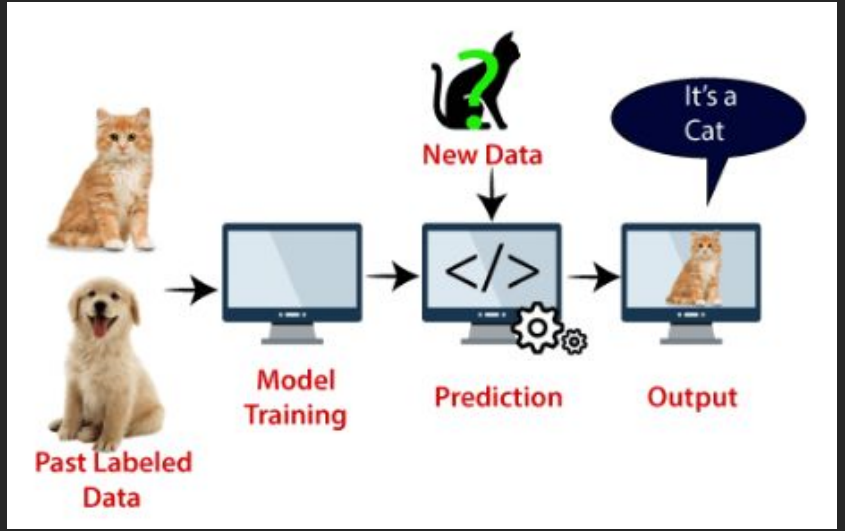
NB with size normalisation and centering



02

SVM

Support Vector Machine

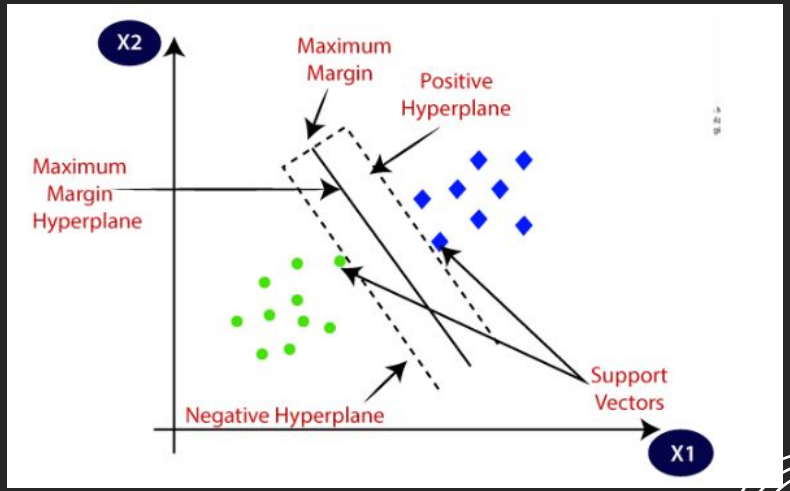


- Provided by MATLAB and other coding languages like Python
- Used for classification and regression problems
- Creates best line-decision boundary - HYPERPLANE
- Classifies n-dimension plane into classes for data point entry
- Uses extreme points/vector - SUPPORT VECTORS

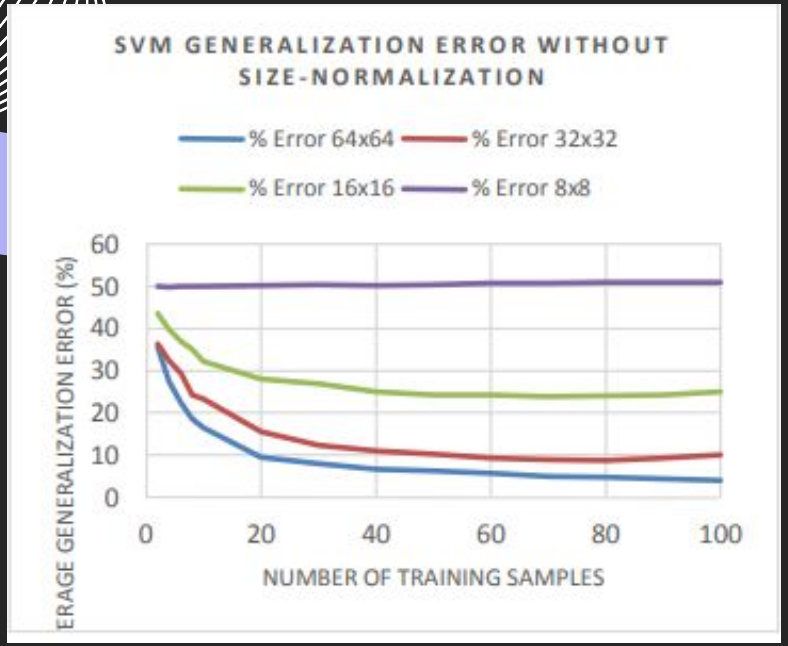
$$\min_{\gamma,\omega,b} \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{m}\xi_i$$

$$s.t.\, y^{(i)}\left(\omega^T x^{(i)} + b\right) \geq 1 - \xi_i, i = 1, \ldots, m$$

$$\xi_i \geq 0, i = 1, \ldots, m$$



SVM with size normalisation



RESOLUTION	GENERALIZATION ERROR (%)
64x64	4
32x32	10
16x16	25
8x8	51

SVM without size normalisation

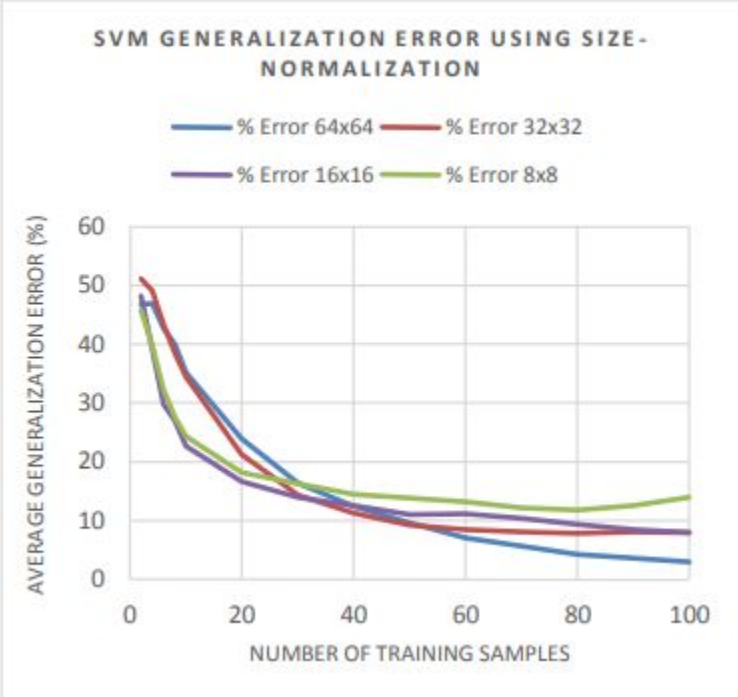
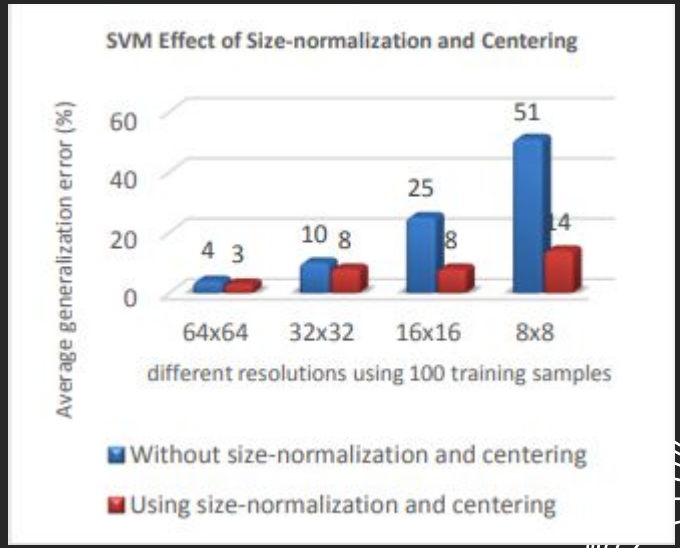


Image Compression

- No control over pixel information
- Leads to loss of information about training and testing database
- Makes it difficult for identification

Normalization helps:

- Center the text and scale it
- Helpful when text is near the edge



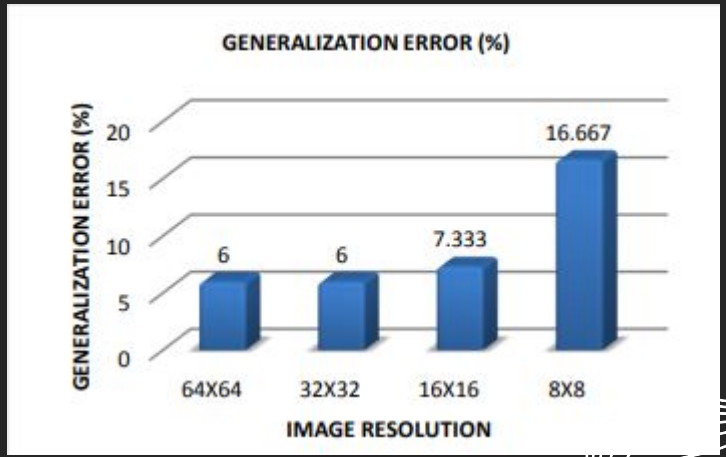
SVM for multiple authors

- Extend to multi-class algorithm
- Compute one class classifier for each class by pitting it against all classes

Example

Take ‘k’ classes
Compute k classifiers

For new sample:
One-against-all approach



Conclusion

Training Sample	Resolution	Use
<10	64x64	SVM
>10	64x64	Naive Baye's without normalization and centering
20-90	8x8	SVM is better but difference is not too much
	32x32 or lower	Naive Baye's without normalization and centering

Future Work

Challenge 1

Signature Matching

- Used by bank/law firms
- Signatures don’t reflect actual handwriting
- They can vary document to document
- Size normalization needs to be modified & extended

Challenge 2

Lower the error

- Increase learning rate
- Lower generalization error for lesser number of samples
- Typically organizations have very few samples

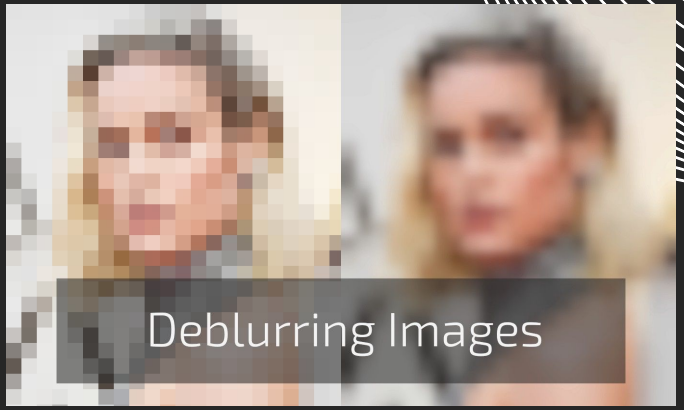
03

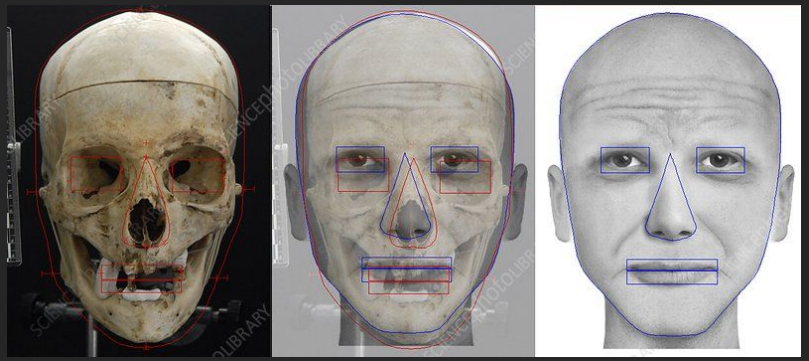
Image reconstruction and deblurring

Introduction

- A target image is a continuous image in the spatial domain
- The digital image is a discrete approximation to the continuous image.
- These images are compressed to save memory storage and also time to compute and transfer.

Concepts covered

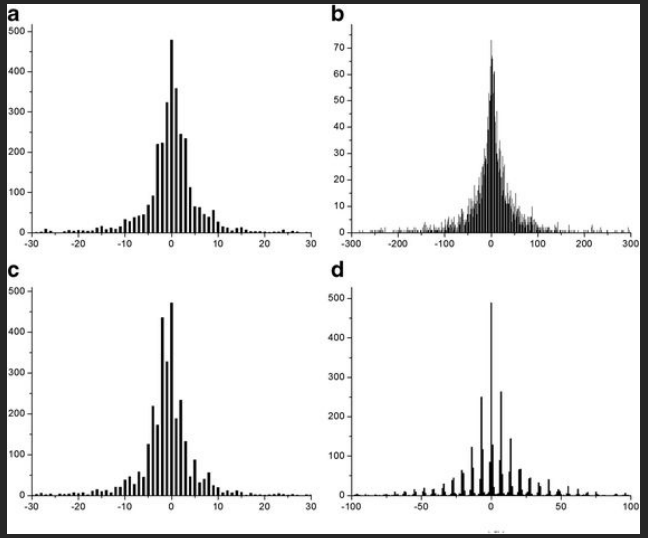


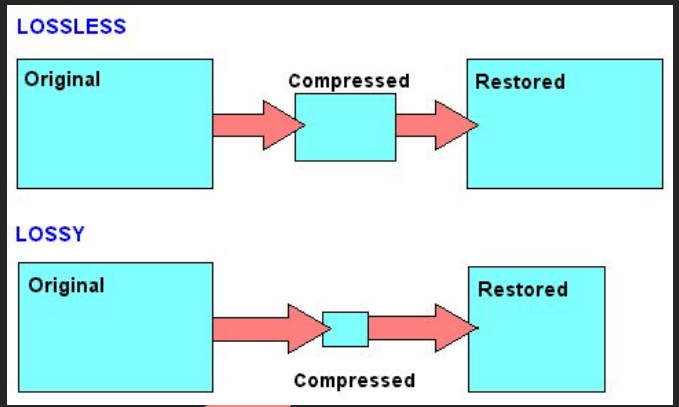


- Discrete cosine transform
- Image compression
- Lossless and lossy algorithms
- Huffman encoding

Algorithms

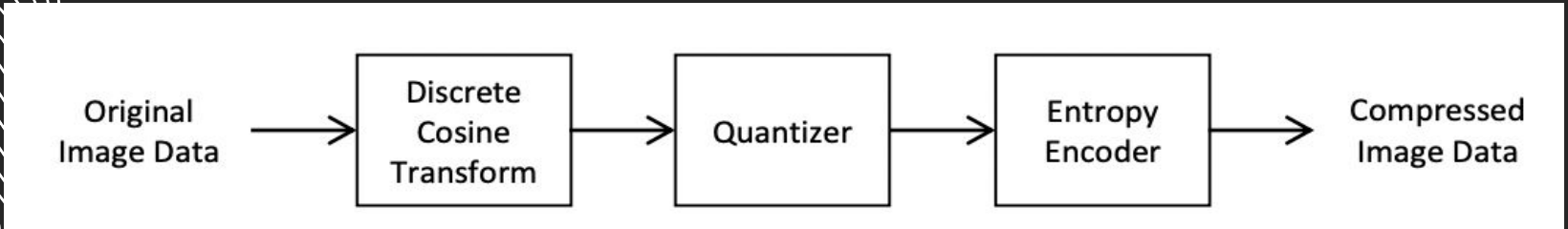
- Lossless- decompressed image contains all the original information
- Lossy-loses some information as it averages or reduces the redundancies



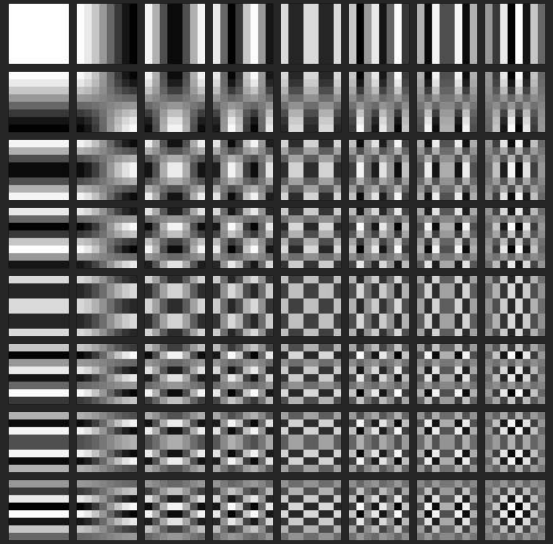


JPEG compression process

- Image compression algorithms used to average the redundancies.



Discrete cosine Transform



Results in coefficients(weights) that will be used to reconstruct the image by superposition of functions

$$F(u,v) = \frac{2}{N} C(u)C(v) \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

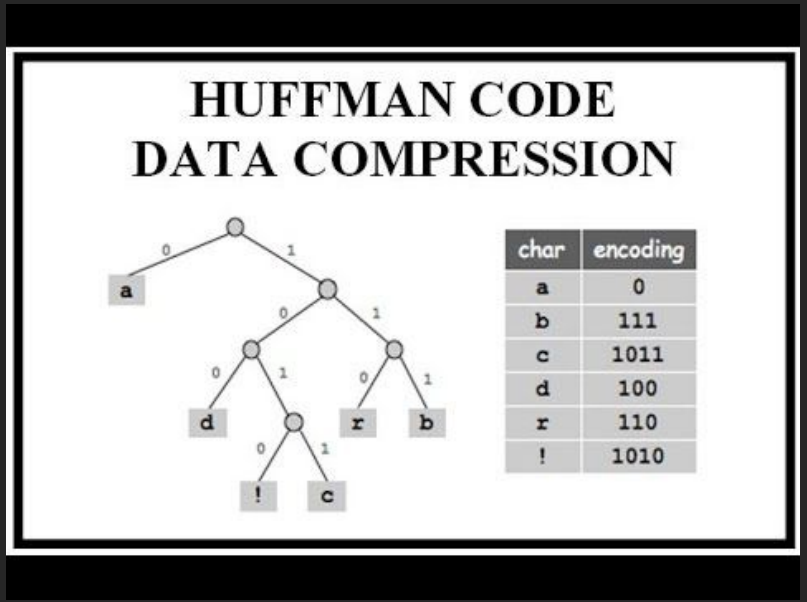
$$f(i,j) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u)C(v)F(u,v) \cos\left[\frac{(2x+1)u\pi}{2N}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right]$$

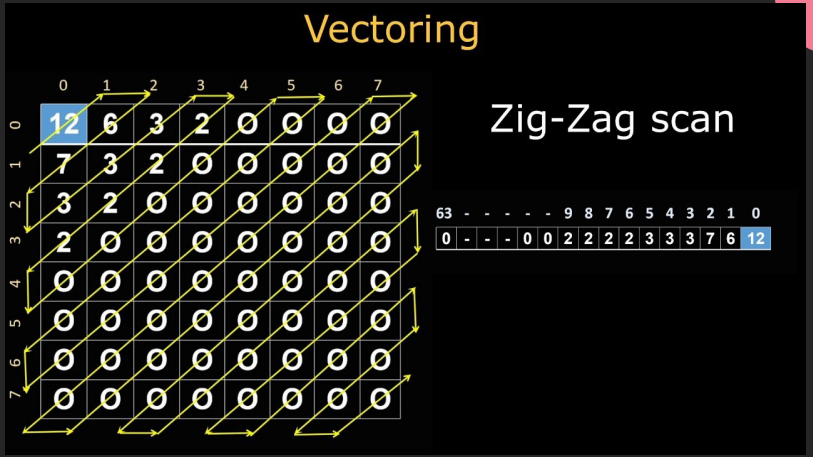
Which can be written in matrix form, where the rows of [T] are the DCT basis vectors, as:

$$[f]_{NxN} = [T^T]_{NxN} [F]_{NxN} [T]_{NxN}$$

$$[F]_{NxN} = [T]_{NxN} [f]_{NxN} [T^T]_{NxN}$$

Huffman encoding





Relevance in forensics

- These images once compressed and decompressed are just an approximation to the original.

- Algorithms aimed at sharpening blurry images rely on understanding the cause of the blur, which may be because of out of focus or captured in motion.

04

The RGB colour model

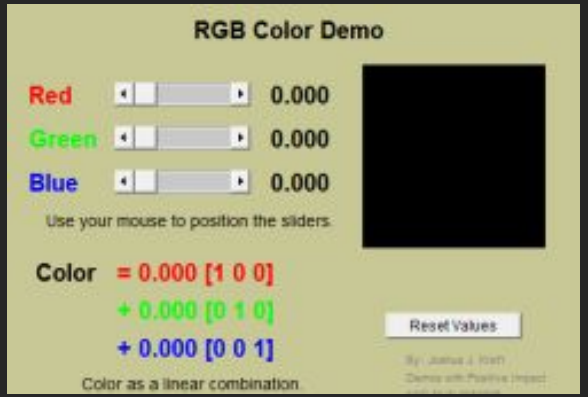
Introduction

- With the ready availability of digital technology, we have all become photographers; and, free, online photo editing tools allow us to alter our pictures to suit our pleasure.
- The downside of this, however, is that when we look at photos, whether in tabloids or on the Web, we wonder: Is this a fake?
- For, just as you can alter pictures to make yourself look more attractive, so too politicians, advertisers, terrorists, and others with particular agendas are using digital imaging tools to manipulate photos to create false impressions.
- Determining whether a photo has been doctored is just one of the many questions that photo forensics tries to answer.

The RGB colour model

-Only three colors ARE needed to create the other colors.

-You can use the sliders to adjust the intensities of each of the Red, Green, and Blue colors to produce new colors that will appear in the large square to the right of the sliders. The intensity of each color ranges between 0 and 1 in increments of 0.005, resulting in 201 possible intensities for each color. Hence, this applet has 3 201 8,120,601 color combinations.



Orange = 0.87 Red + 0.60 Green + 0.26 Blue

Translation

-First, with black and white images, each pixel is colored either black (0 = black) or white (1 = white). To produce images on a gray-scale, each pixel has an intensity value between 0 and 1; in the grayscale, for example, a pixel with intensity value .2 is darker than a pixel with intensity value .6

-With color images, on the other hand, each pixel is assigned three numbers, each between 0 and 1, which represent the intensity values of Red, Green, and Blue, respectively.

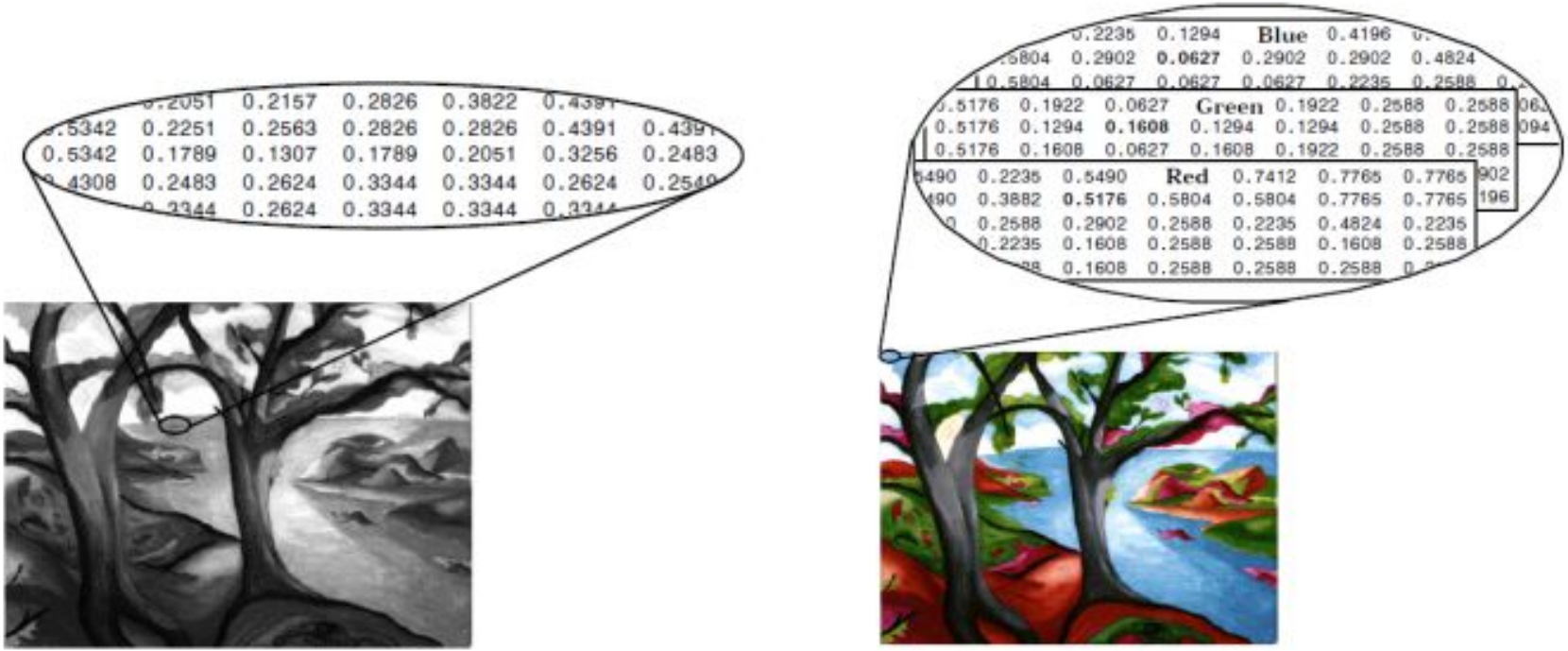


Figure 7. Grayscale and RGB color model images with pixel data

Linear Combination

-Every color in the spectrum can be expressed as a weighted sum of the three primary colors Red, Green, and Blue.

-Because every color in the spectrum is a sum of scalar multiples of Red, Green, and Blue, we say that every color is a linear combination of the three colors Red, Green, and Blue.

2D-DCT

$$C(i,j) = a(i)a(j)\sum_{x=0}^{7}\sum_{y=0}^{7} f(x,y)\cos\left[\frac{(2x+1)i\pi}{16}\right]\cos\left[\frac{(2y+1)j\pi}{16}\right] \text{ for } i,j = 0,1,\ldots 7$$

where

$$a(i) = a(j) = \begin{cases} \sqrt{\frac{1}{8}} & \text{if } i,j = 0 \\ \sqrt{\frac{1}{4}} & \text{otherwise} \end{cases}$$

The 2D-DCT takes a discrete set of 8 x 8 = 64 points in the spatial domain (x,y) and converts them to the frequency domain (i,j). The resulting coefficients C(i,j) are the weights that will be used to reconstruct the image using the 2D-ICDT

2D-IDCT

$$f(x,y)=\sum_{i=0}^{7}\sum_{j=0}^{7}C(i,j)a(i)a(j)\cos\left[\frac{(2x+1)i\pi}{16}\right]\cos\left[\frac{(2y+1)j\pi}{16}\right]\text{ for } x,y=0,1,\ldots 7$$

-In the case of JPEG compression, M = N = 8 and there are 64 resulting basis functions corresponding to

$$a(i)a(j)\cos\left[\frac{(2x+1)i\pi}{16}\right]\cos\left[\frac{(2y+1)j\pi}{16}\right]\text{ for } i,j=0,1,\ldots 7$$

-Hence, the recovered image f(x,y) is a linear combination of these functions with weights C(i,j)

-These 64 basis functions are the discrete patterns given BELOW

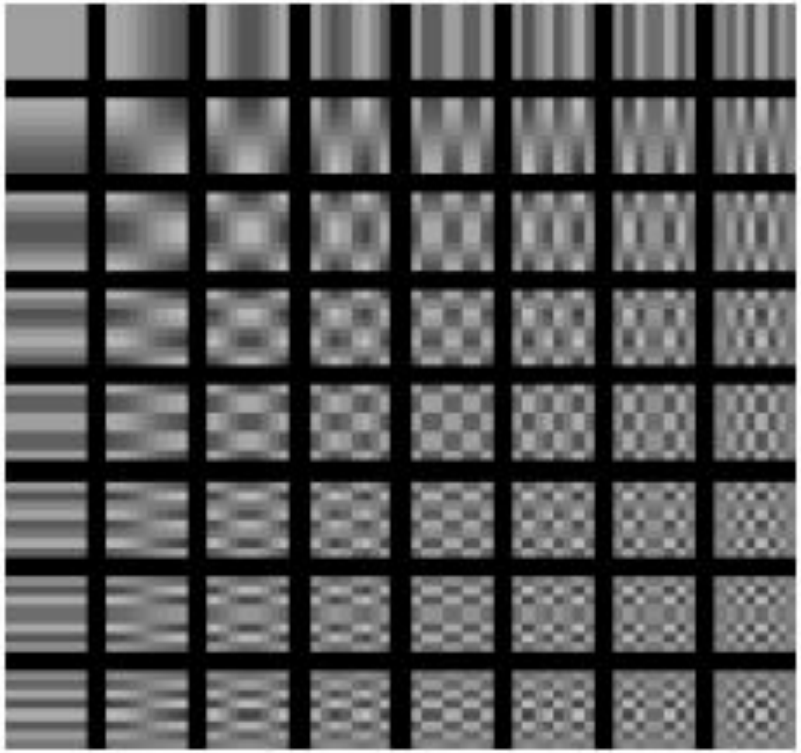


Figure 10. Graphical representation of the 2D-DCT for N = 8 (8 x 8 blocks)

Let's do a sum, shall we?

Example of IDCT(**Inverse Discrete Cosine Transform**). Start with an 8 x 8 pixel image of the capital letter A

The 2D-DCT coefficients are computed for the image using the equations above and given in the matrix

6.1917	−0.3411	1.2418	0.1492	0.1583	0.2742	−0.0724	0.0561
0.2205	0.0214	0.4503	0.3947	−0.7846	−0.4391	0.1001	−0.2554
1.0423	0.2214	−1.0017	−0.2720	0.0789	−0.1952	0.2801	0.4713
−0.2340	−0.0392	−0.2617	−0.2866	0.6351	0.3501	−0.1433	0.3550
0.2750	0.0226	0.1229	0.2183	−0.2583	−0.0742	−0.2042	−0.5906
0.0653	0.0428	−0.4721	−0.2905	0.4745	0.2875	−0.0284	−0.1311
0.3169	0.0541	−0.1033	−0.0225	−0.0056	0.1017	−0.1650	−0.1500
−0.2970	−0.0627	0.1960	0.0644	−0.1136	−0.1031	0.1887	0.1444

The final image of the letter A is reconstructed as the Inverse DCT and shown in Figure below. This final image is essentially a linear combination of the basis functions with their associated DCT coefficients as the weights. Each figure contains three images: The image on the right is the DCT basis function along with the computed DCT coefficient. The middle image shows what the basis function multiplied by the associated weight looks like. The left image shows the resulting linear combination using up to 10, 36, and all 64 basis functions.

Catching Criminals with frequent incidents

- Dr. Kim Rossmo holds the University Chair in Criminology and is the Director of the Center for Geospatial Intelligence and Investigation in the School of Criminal Justice and Criminology at Texas State University

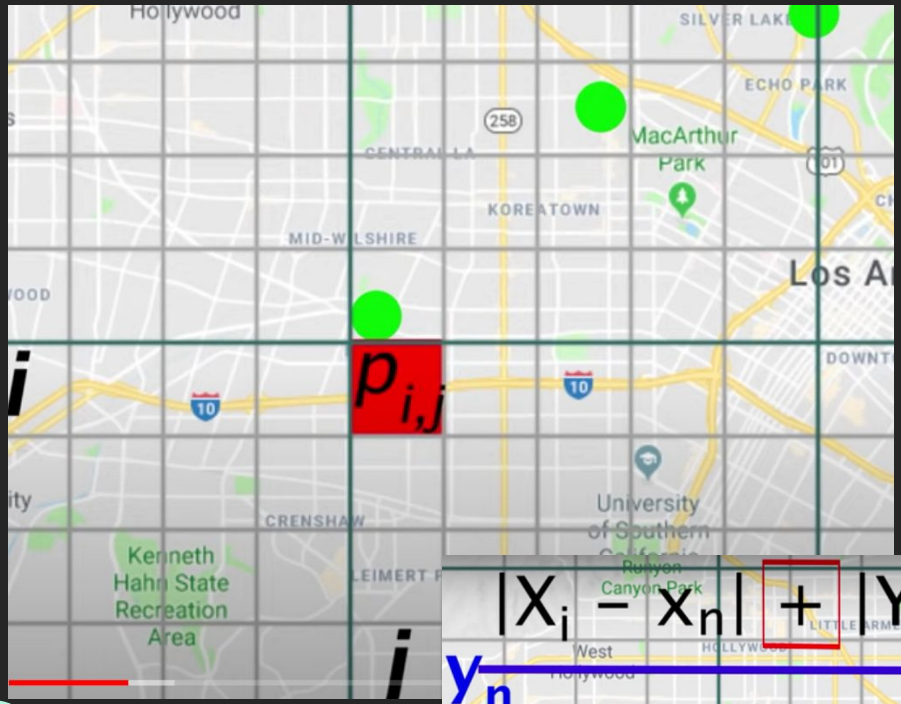
- He tried to find a formula that could find where the criminal exactly lives

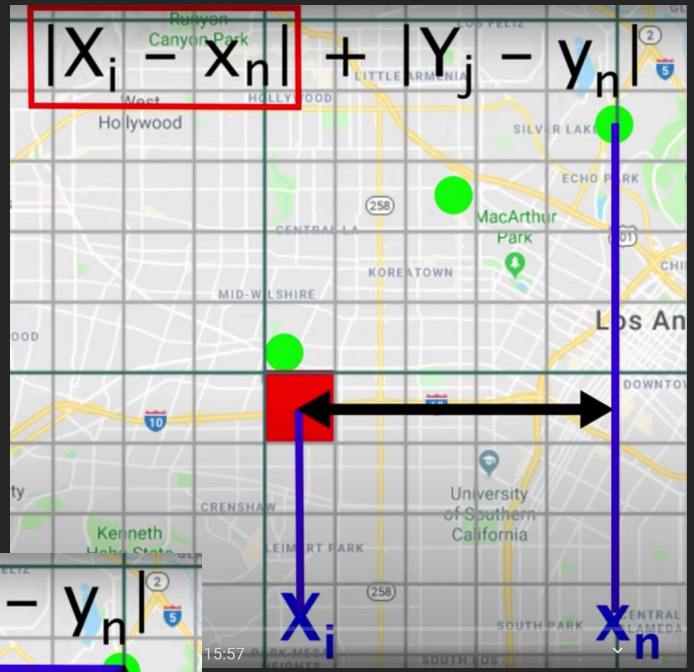
- Based on past data, he know that criminals don't often commit crimes right by their own

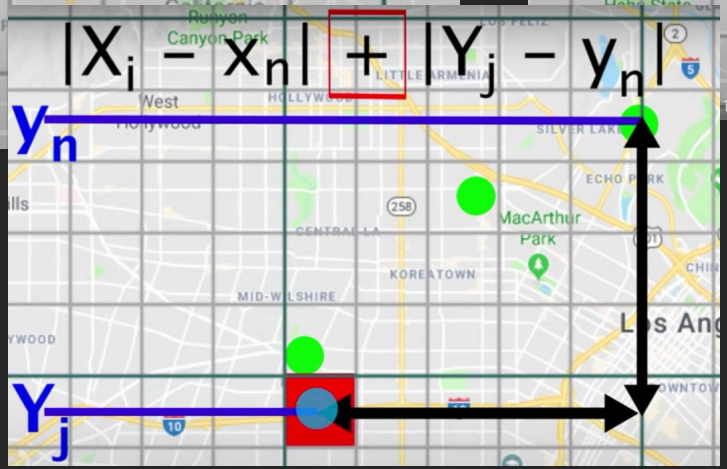
Home but they don't essentially move too far off so we can determine a "HOT ZONE" to find

Their exact location.

$$p_{i,j} = k \sum_{n=1}^{(\text{total crimes})} \left[\underbrace{\frac{\phi_{ij}}{(|X_i - x_n| + |Y_j - y_n|)^f}}_{1^{\text{st}}\ \text{term}} + \underbrace{\frac{(1-\phi_{ij})(B^{g-f})}{(2B - |X_i - x_n| - |Y_j - y_n|)^g}}_{2^{\text{nd}}\ \text{term}} \right]$$

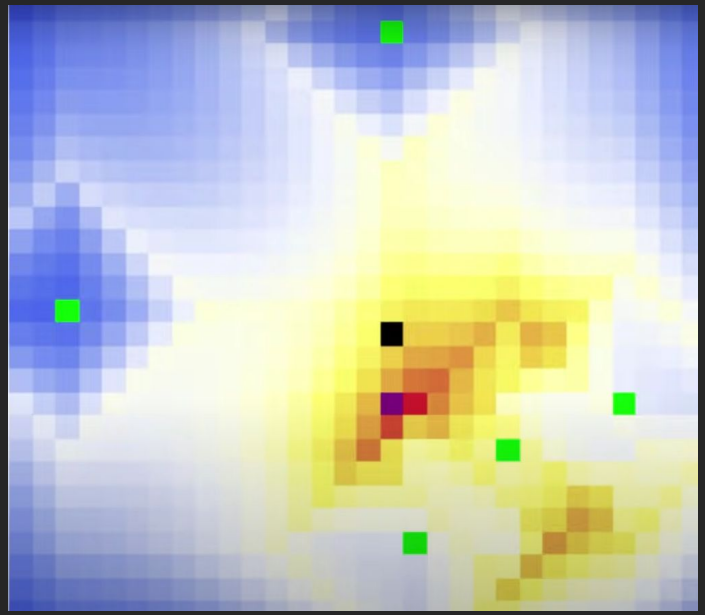

Hollywood
SILVER LAKE
ECHO PARK
MacArthur Park
CENTRAL LA
KOREATOWN
MID-WILSHIRE
Los A
DOWNT
$p_{i,j}$
i
i
University of Southern
CRENSHAW
Kenneth Hahn State Recreation Area

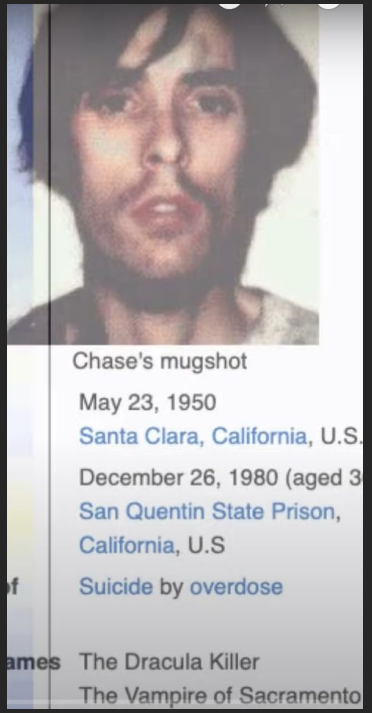

$|X_i - x_n| + |Y_j - y_n|$
Runyon Canyon Park
West Hollywood
MacArthur Park
KOREATOWN
MID-WILSHIRE
University of Southern California
CRENSHAW
Kenneth
X_i
x_n
15:57


$|X_i - x_n| + |Y_j - y_n|$
Runyon Canyon Park
West Hollywood
y_n
MacArthur Park
KOREATOWN
MID-WILSHIRE
Y_j

Richard Chase- a serial killer

Do this for every square in our grid and we generate a heat map





Thank You!

Do you have any questions?