Work Experience

Kai Chen

Resume

- Oct 2018 - present, Data Science Manager, Unitymedia, Germany
- May 2017 - Oct 2018, Senior Data Scientist, AGT International, Germany
- Sep 2012 - Jun 2017, PhD in CS, University of Fribourg, Switzerland
- Jun 2015 - Sep 2015, Visiting PhD, Chinese Academy of Sciences, China
- Sep 2009 - Sep 2012, Master in CS, University of Fribourg, Switzerland
- Oct 2005 - Feb 2008, Bachelor in CS, University of Applied Science, Switzerland

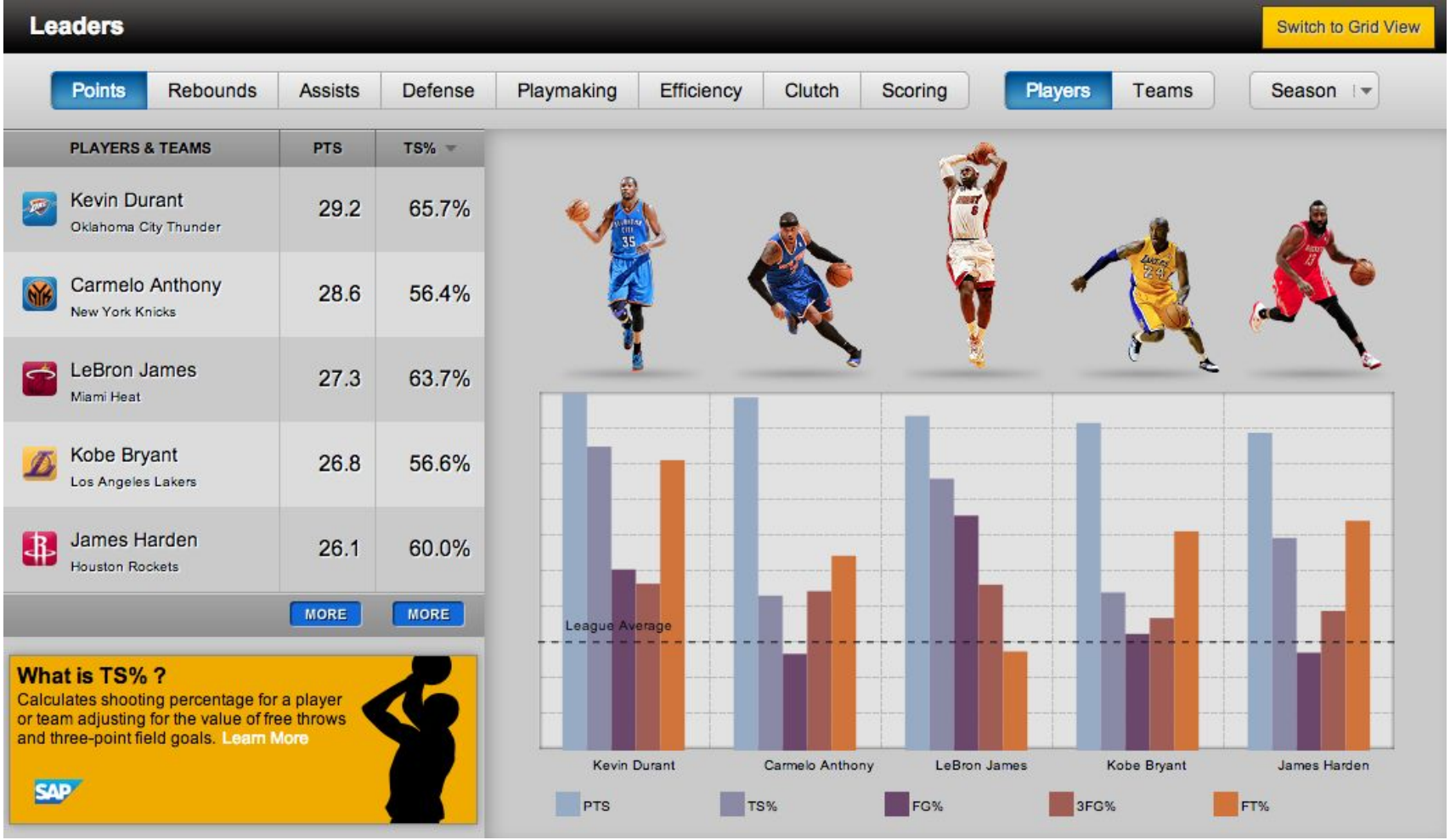
Github: github.com/ck-unifr
Google scholar: kai chen unifr
Linkedin: linkedin.com/in/kai-chen-29503288/

Work at Unitymedia

- Recommendation System
- Churn Prevention
- ETL in Hadoop

Work at AGT International

- Punch Recognition with Deep Learning
- A/B Testing for Punch Recognition Evaluation
- Anomaly Detection



The NBA brings Big Data at your fingertips
http://labs.sogeti.com/the-nba-brings-big-data-at-your-fingertips/

UFC 196 | SAT. MAR. 5, 2016 | MCGREGOR VS. DIAZ

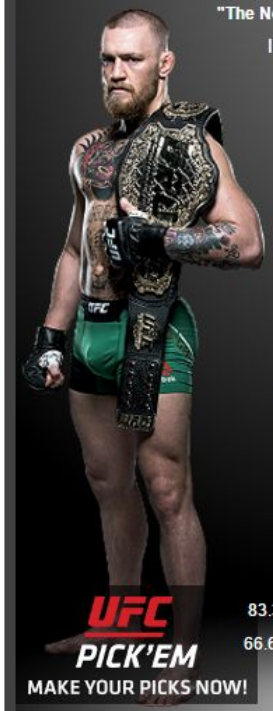
10PM/7PM ETPT | Las Vegas, Nevada

FANTASY | TICKETS | HOW TO WATCH | PROGRAM | VIDEOS | NEWS | PRINT

FIGHT CARD | LIVE STATS | ODDS

MAIN CARD

Conor McGregor "The Notorious"		#5 Nate Diaz
IRELAND	COUNTRY	USA
19-2-0	RECORD	19-10-0
89%	KO/TKO	26%
5%	SUB	58%
5%	DEC	16%
69 in	HEIGHT	70 in
170 lbs	WEIGHT	170 lbs
74 in	REACH	76 in
40 in	LEG REACH	38 in
	SIGNIFICANT STRIKES	
5.53	LANDED PER MINUTE	4.3
44.72%	ACCURACY	43.24%
3.02	ABSORBED P/M	3.29
64.24%	DEFENSE	56.7%
	GRAPPLING	
1.92	TAKEDOWN AVERAGE	1.21
83.33%	TAKEDOWN ACCURACY	30.77%
66.67%	TAKEDOWNS DEFENDED	44.44%
000	SUBMISSION AVERAGE	1.51



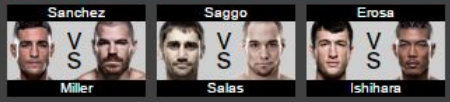
UFC PICK'EM

MAKE YOUR PICKS NOW!

MAIN CARD — 10PM/7PM ETPT

FS1 PRELIMS — 8PM/5PM ETPT

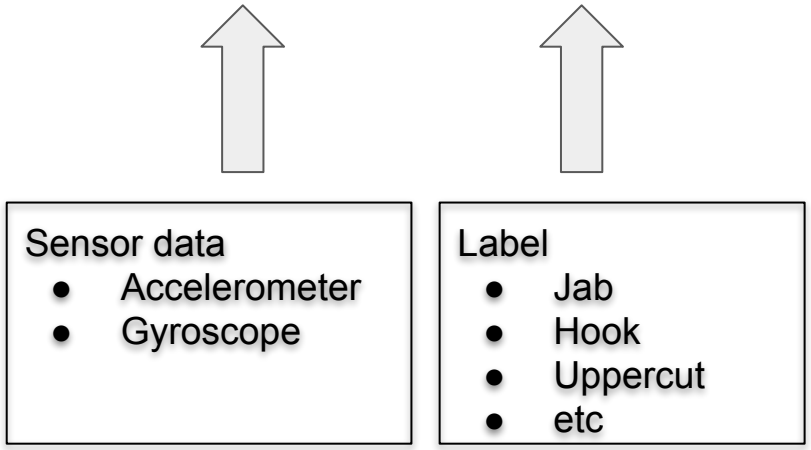
UFC FIGHT PASS EARLY PRELIMS — 6:30PM/3:30PM ETPT



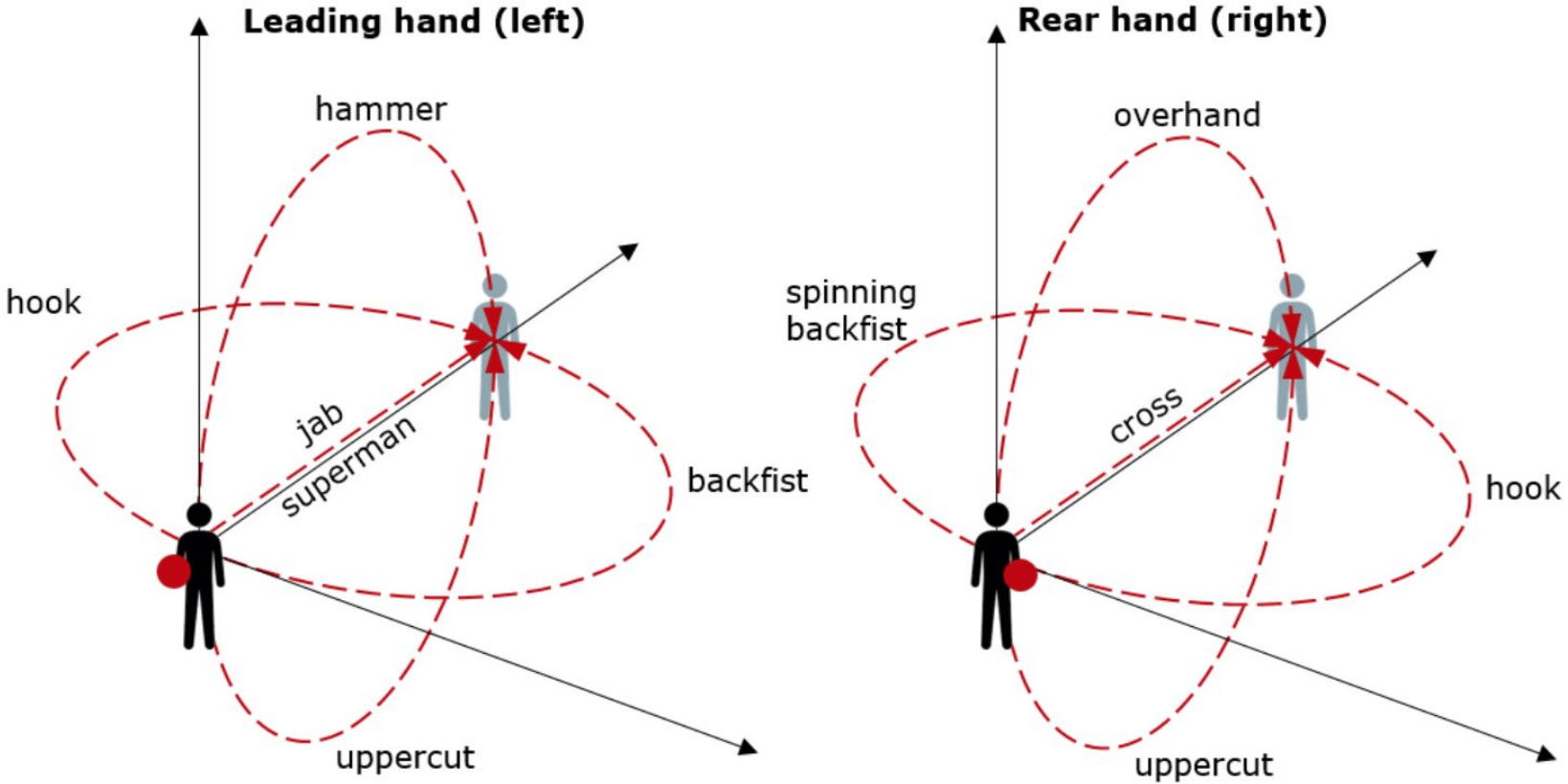
*FIGHTS ARE VERBALLY AGREED UPON AND CARD MAY CHANGE AT ANY TIME. SOME FIGHTS MAY NOT BE BROADCAST.

Objective

$$f : X \mapsto Y$$

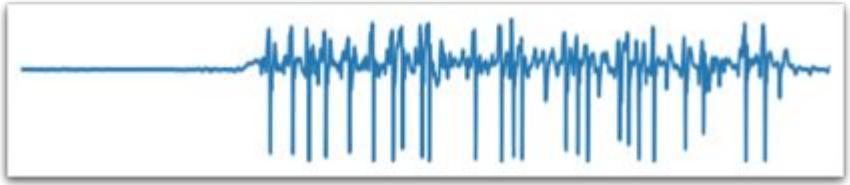


Punch


Leading hand (left)
hammer
hook
jab
superman
backfist
uppercut
Rear hand (right)
overhand
spinning
backfist
cross
hook
uppercut

Why Challenging?

What we see



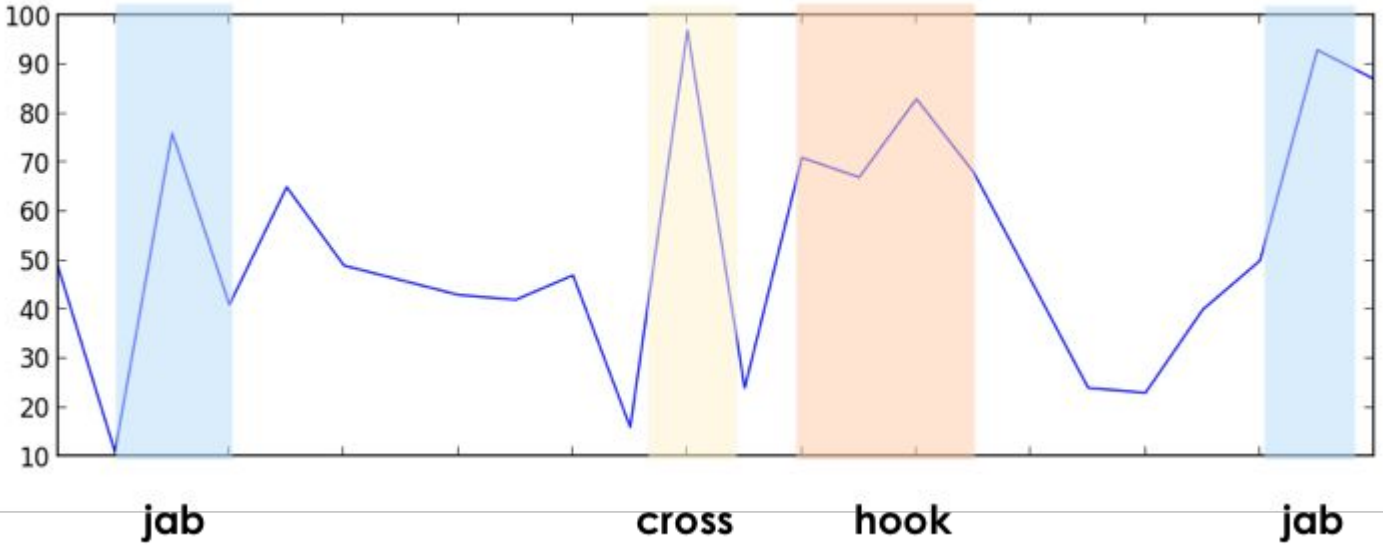
What a computer see

Why Machine Learning?

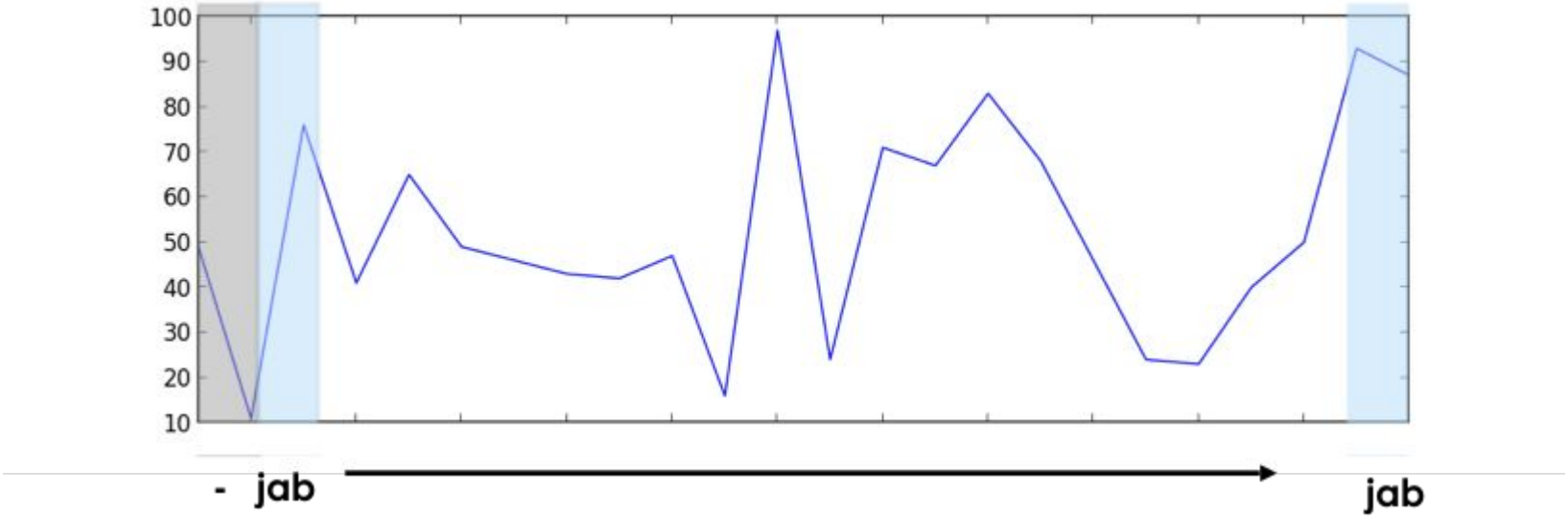
- We cannot code the rules
- We cannot scale

Amazon Machine Learning,
docs.aws.amazon.com/machine-learning/latest/dg/when-to-use-machine-learning.html

Train



Predict



Why Deep Learning?

ImageNet Challenge

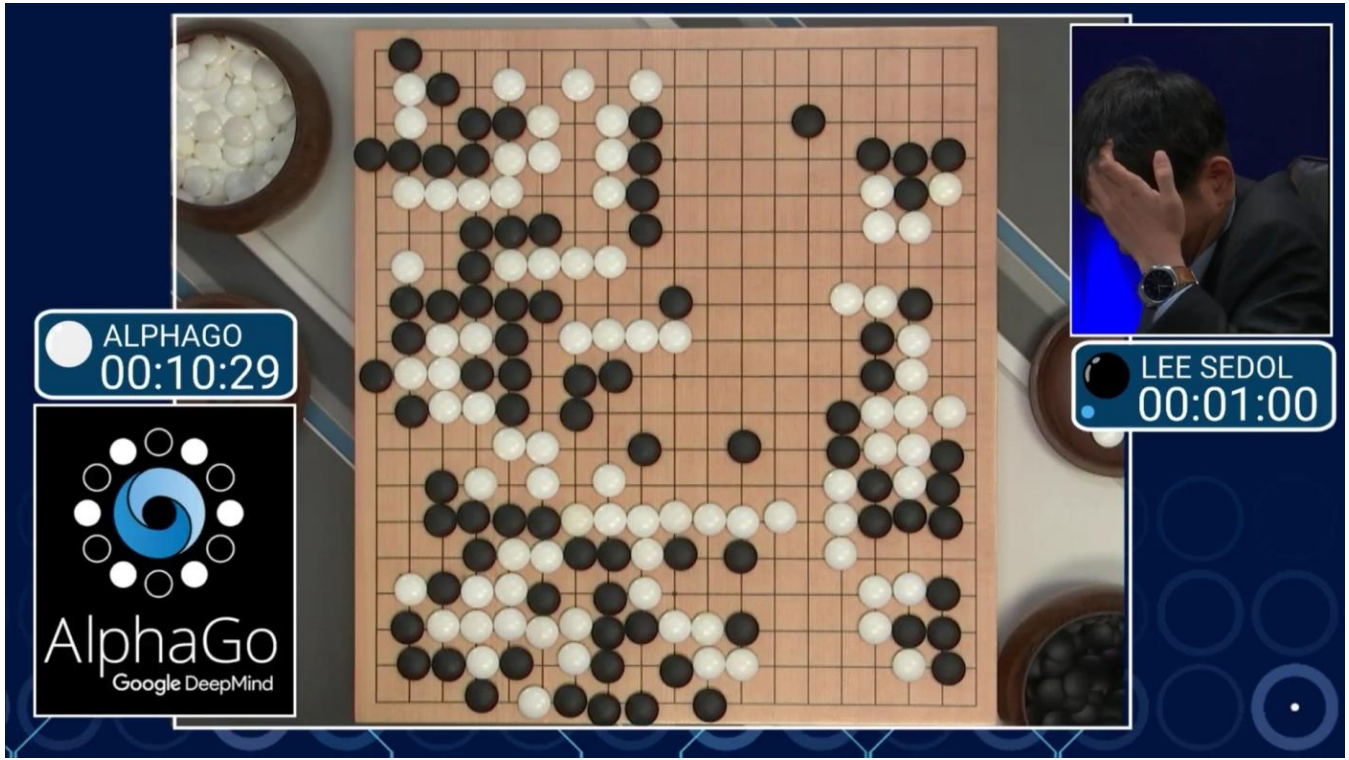
IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.

4

AlexNet [Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever 2012]

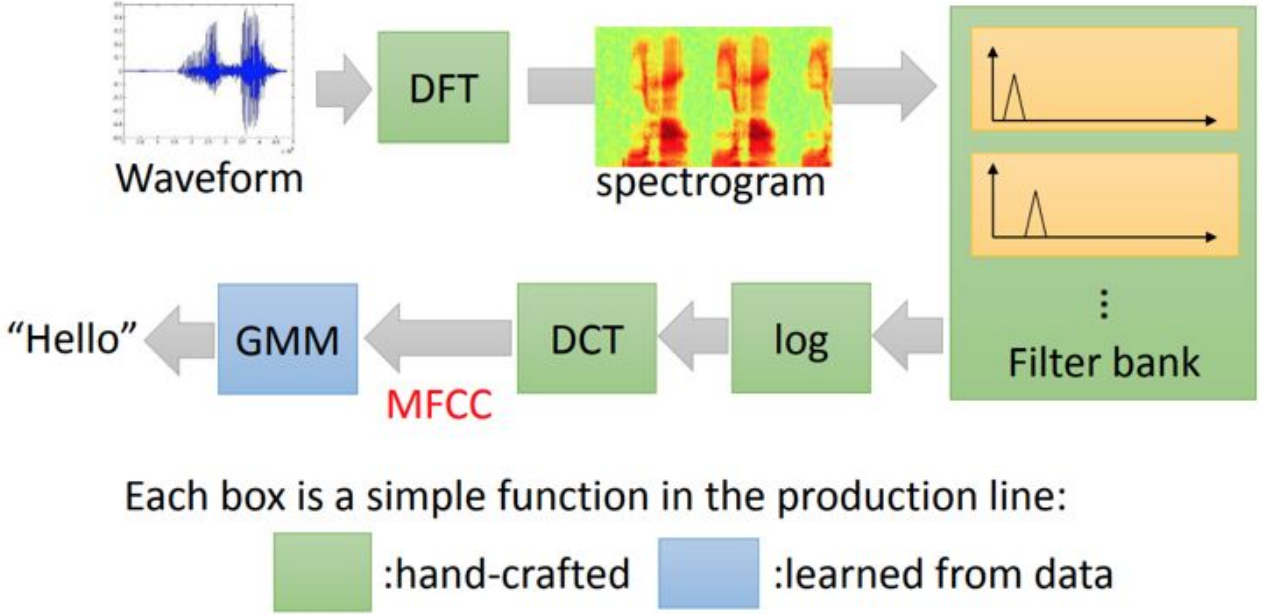
Why Deep Learning?



AlphaGo [DeepMind 2016]

Why Deep Learning?

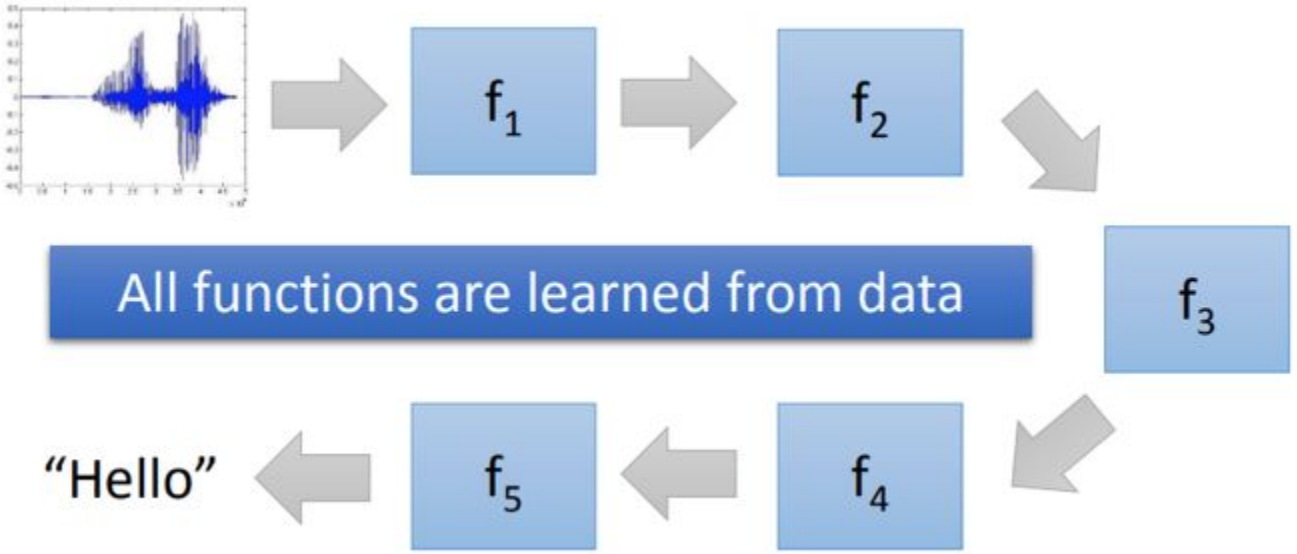
- Shallow Learning



Source: Hung-yi Lee, Machine Learning and having it deep and structured, National Taiwan University , 2015.

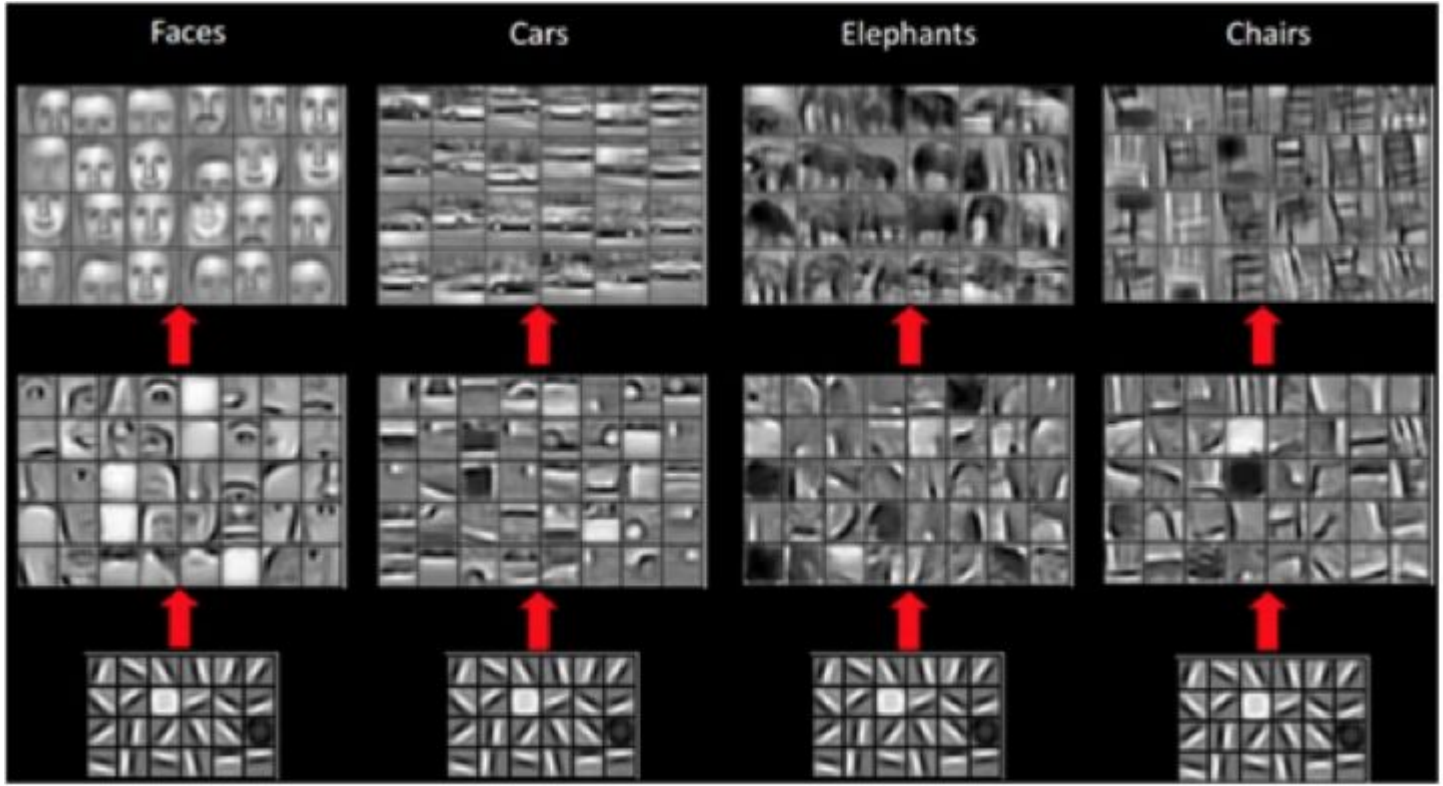
Why Deep Learning?

- Deep Learning

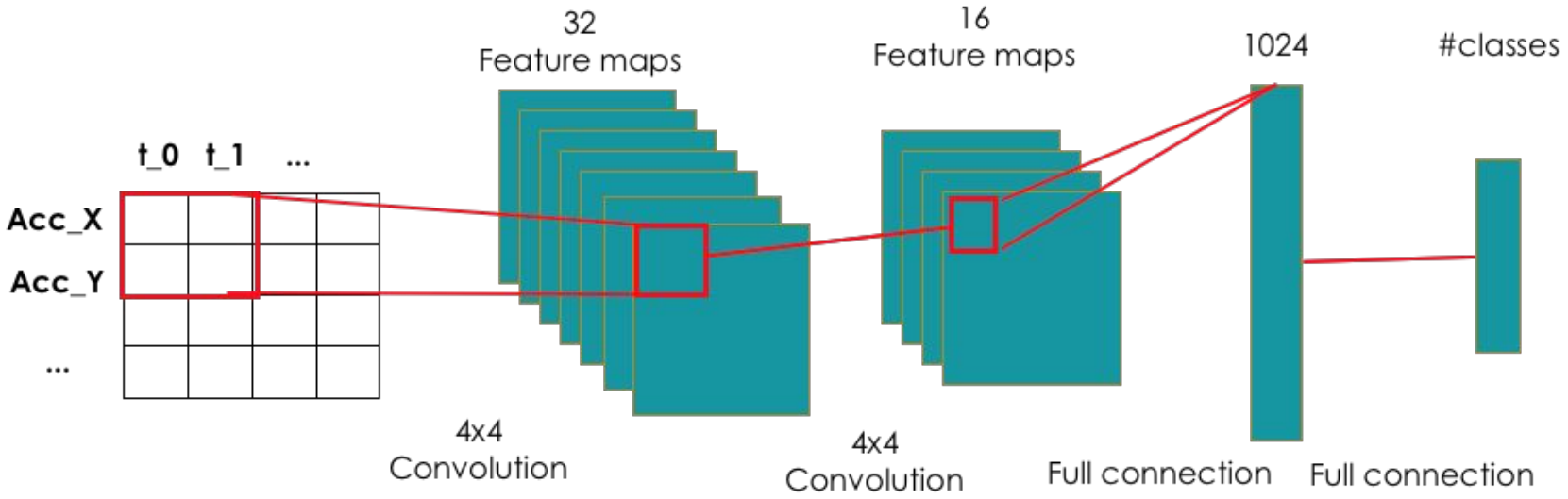


"Bye bye, MFCC, Deng Li, Interspeech, 2014
Source: Hung-yi Lee, Machine Learning and having it deep and structured, National Taiwan University , 2015.

Convolutional Neural Networks (CNN)



CNN for Punch Recognition



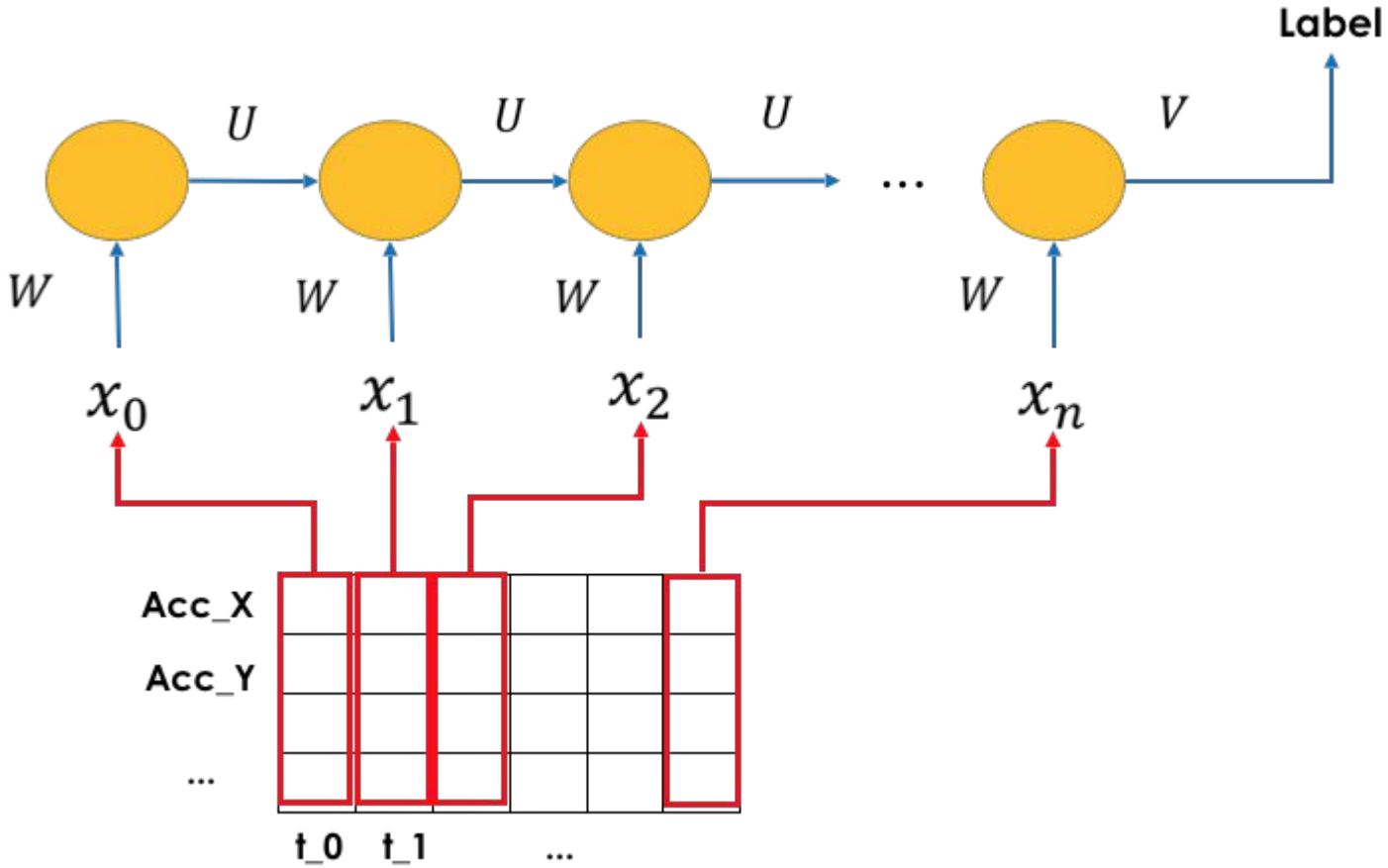
- Batch normalization
- Activation (Conv Layers): ReLU
- Strides: 1, 1
- Kernel initialization: He normal initializer, (He et al., http://arxiv.org/abs/1502.01852)
- Dropout: 0.2
- Optimizer: RMSProp

Recurrent Neural Network (RNN)

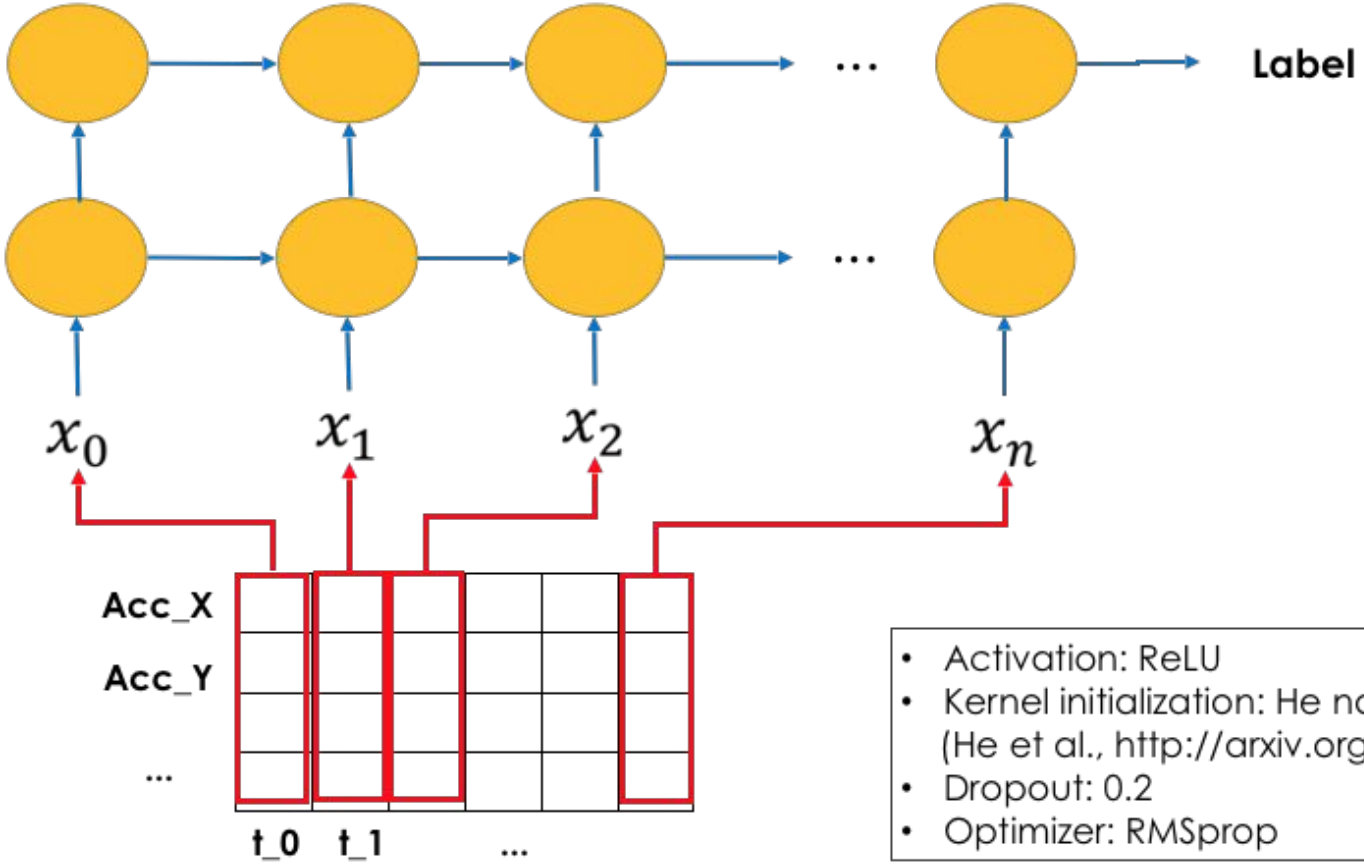
“ The food was bad, not good at all.”

“ The food was good, not bad at all.”

RNN for Punch Recognition

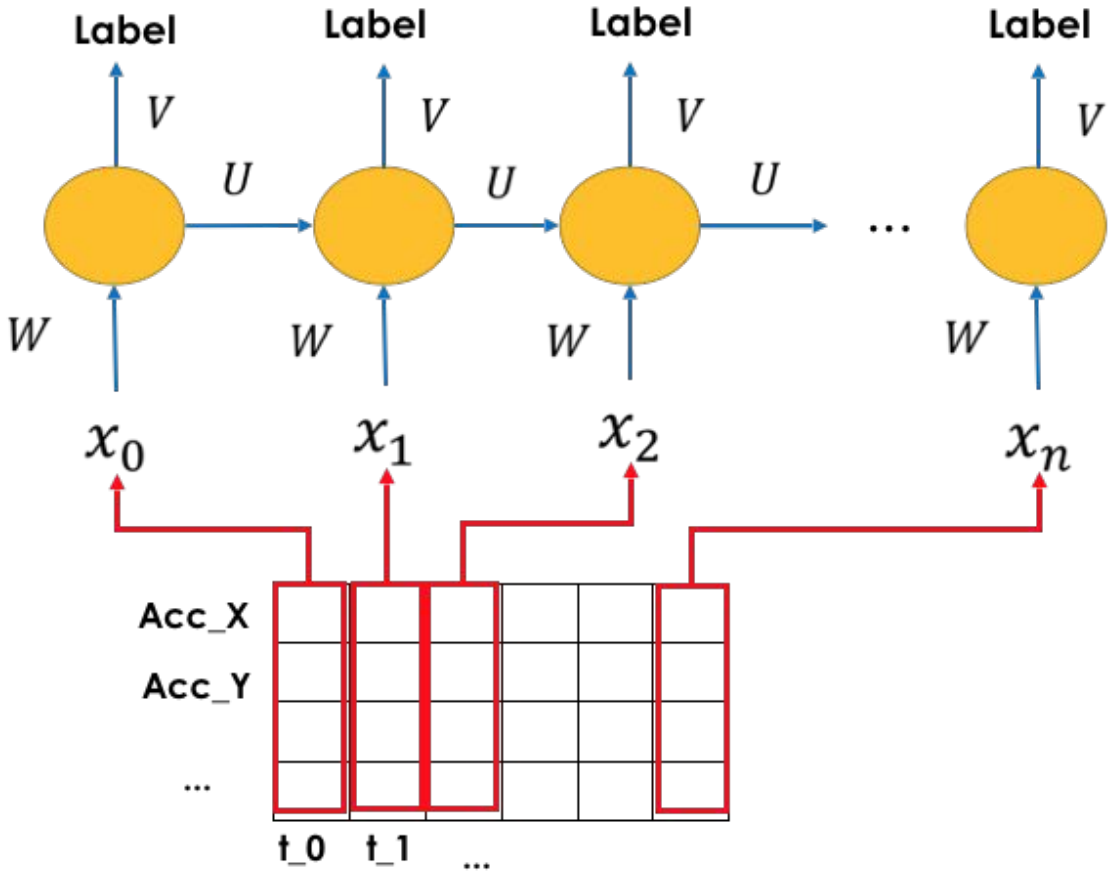


RNN for Punch Recognition

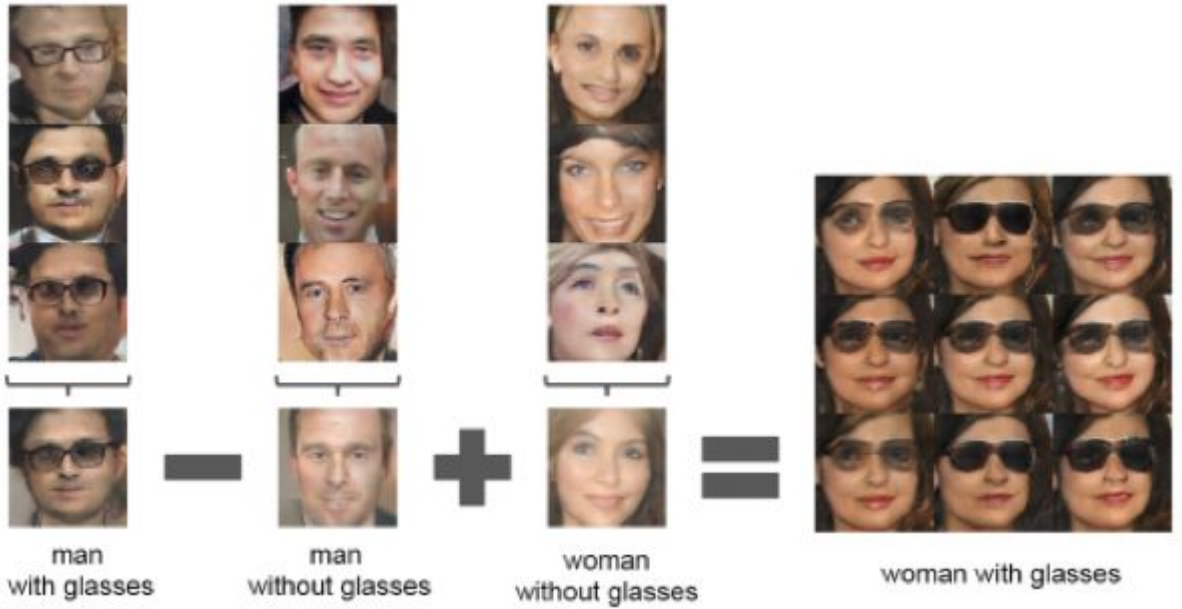


- Activation: ReLU
- Kernel initialization: He normal initializer, (He et al., http://arxiv.org/abs/1502.01852)
- Dropout: 0.2
- Optimizer: RMSprop

RNN for Punch Recognition



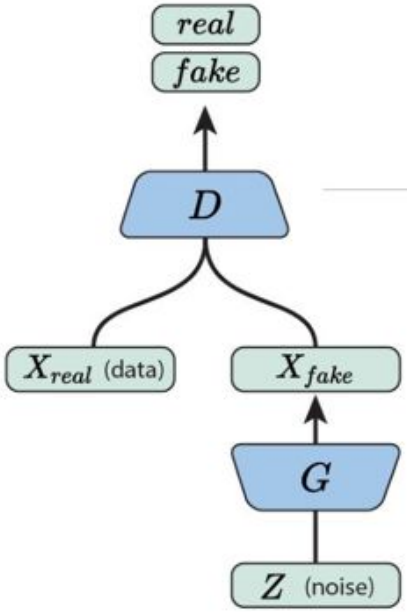
Generative Adversarial Networks (GANs)



Radford et al, „Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks“, arXiv:1511.06434
github.com/Newmu/dcgan_code

GANs

Generative Adversarial Networks (GANs) are a way to make a generative model by having two neural networks compete with each other.



The **discriminator** tries to distinguish genuine data from forgeries created by the generator.

The **generator** turns random noise into immitations of the data, in an attempt to fool the discriminator.

Chris Olah 2016

Future Work

- Combine sensor and image data for punch recognition
- Transfer learning

Anomaly Detection in Sports

Anomaly Detection

- From the perspective of the audience
  - Score
  - Turnover
  - Foul out
  - Rejection
  - Injury
  - ...

Anomaly Detection

- From the perspective of a programmer
  - Unsupervised learning
    - **Outlier detection**
  - Supervised learning
    - Neural networks
    - Gradient boost machine
    - ...

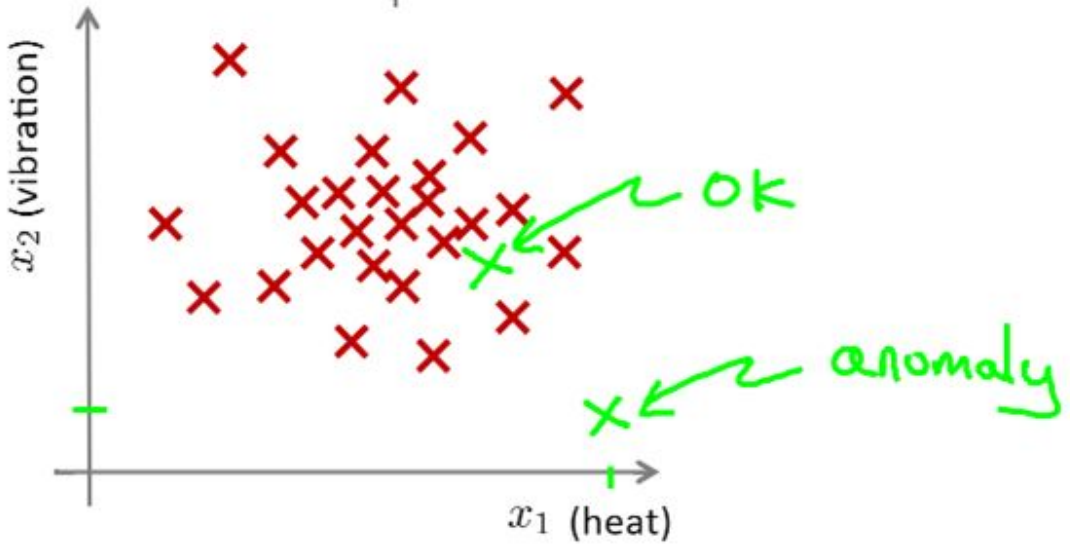
Anomaly detection example

Aircraft engine features:

- x_1 = heat generated
- x_2 = vibration intensity
- ...

Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$

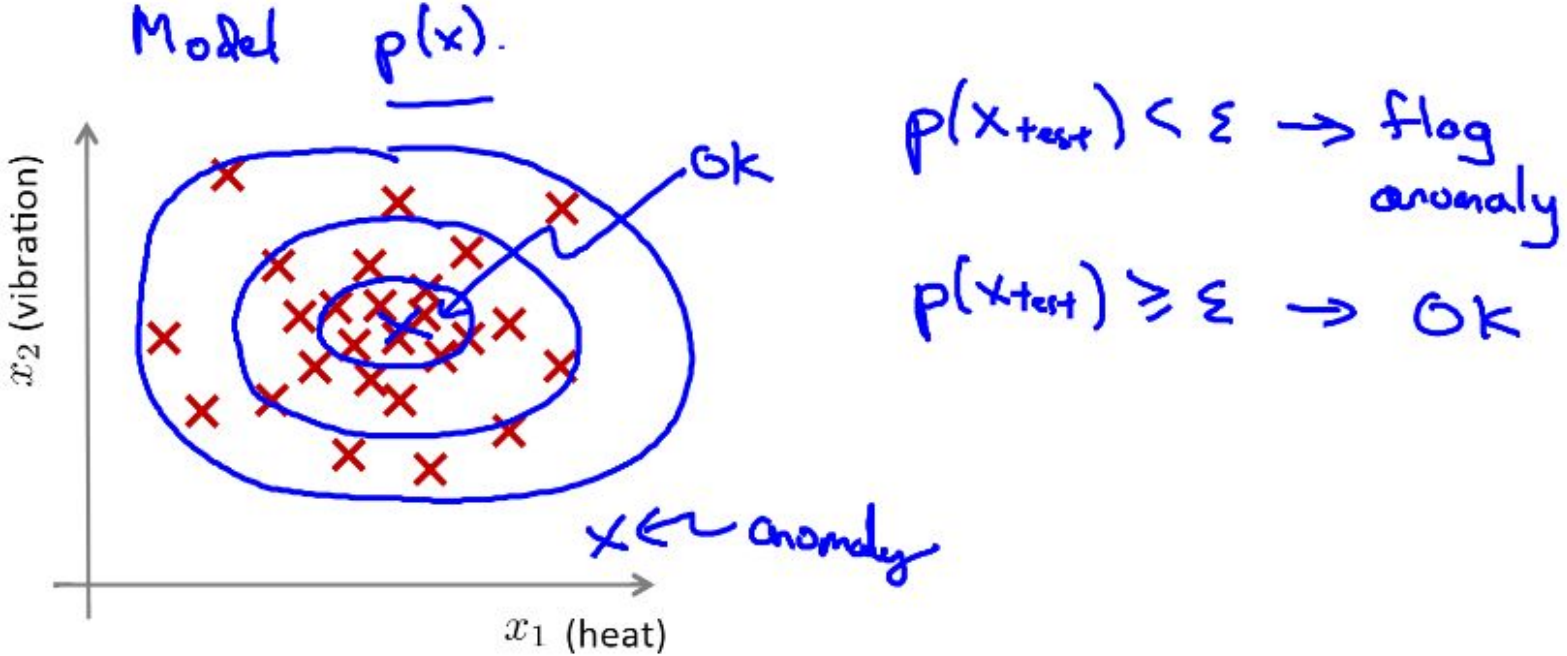
New engine: x_{test}



Andrew Ng

Density estimation

- Dataset: $\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$
- Is x_{test} anomalous?



Andrew Ng, Machine Learning, week 9, Lecture 15, Coursera.

Anomaly detection example

- Fraud detection:
  - $x^{(i)}$ = features of user i's activities
  - Model $p(x)$ from data.
  - Identify unusual users by checking which have $p(x) < \varepsilon$
- Manufacturing
- Monitoring computers in a data center.
  - $x^{(i)}$ = features of machine i

  x_1 = memory use, x_2 = number of disk accesses/sec,

  x_3 = CPU load, x_4 = CPU load/network traffic.

  ...

x_1 x_2 x_3 x_4 $p(x)$

$p(x) < \varepsilon$

Methodology

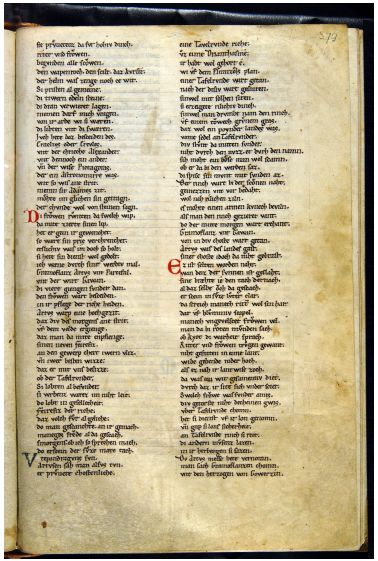
- Robust Covariance [1]
- Isolation Forest [1]
- Local Outlier Factor [1]
- One-class SVM [1]
- Gaussian Mixture Model [2]

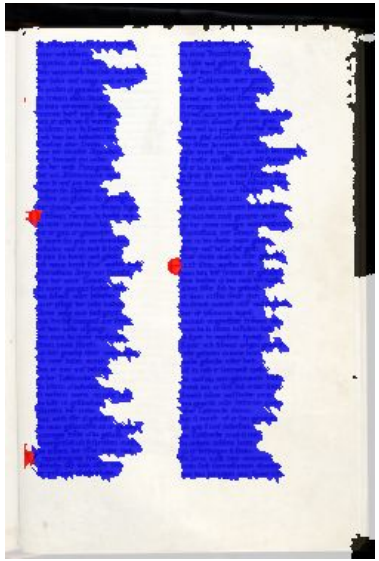
[1] scikit-learn, Novelty and Outlier Detection. http://scikit-learn.org/stable/modules/outlier_detection.html
[2] Andrew Ng, Anomaly Detection, Coursera. https://www.coursera.org/learn/machine-learning/home/week/9

Historical Document Layout Analysis with Machine Learning

Introduction

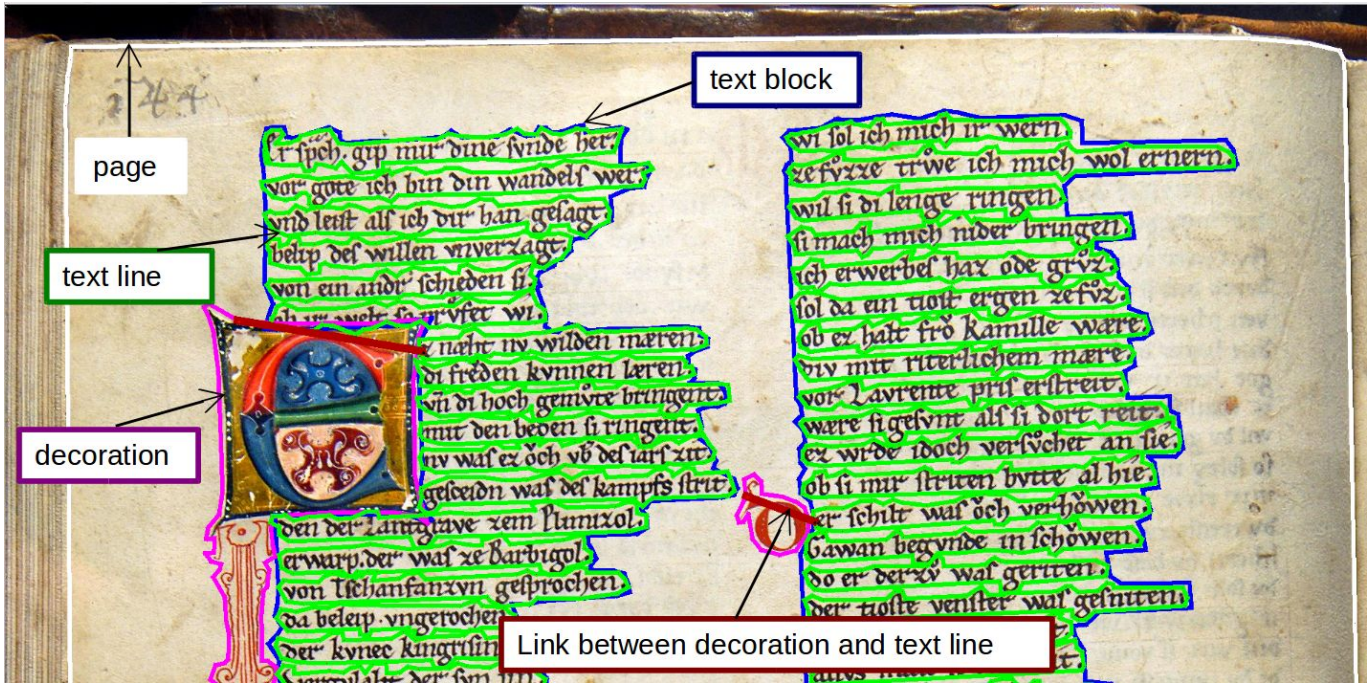
- Goal
  - Developing a general page segmentation method with minimal prior knowledge.
- Basic Idea
  - Page Segmentation ⟶ Pixel Labeling



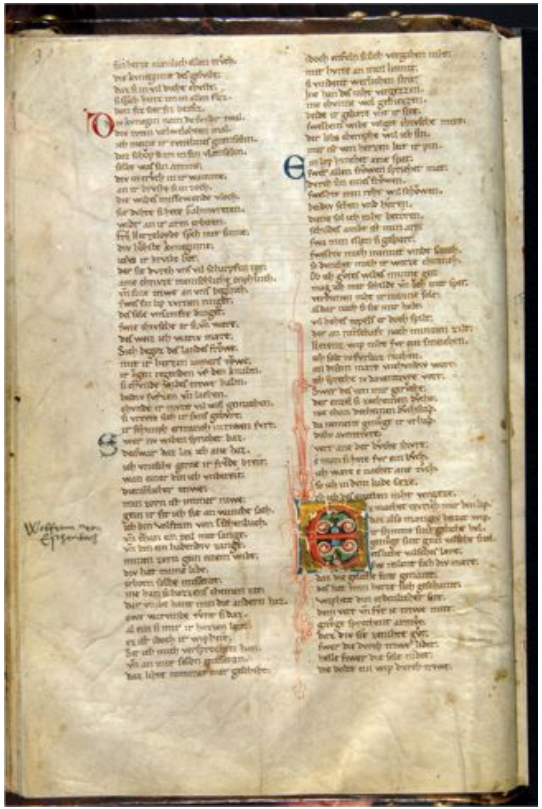


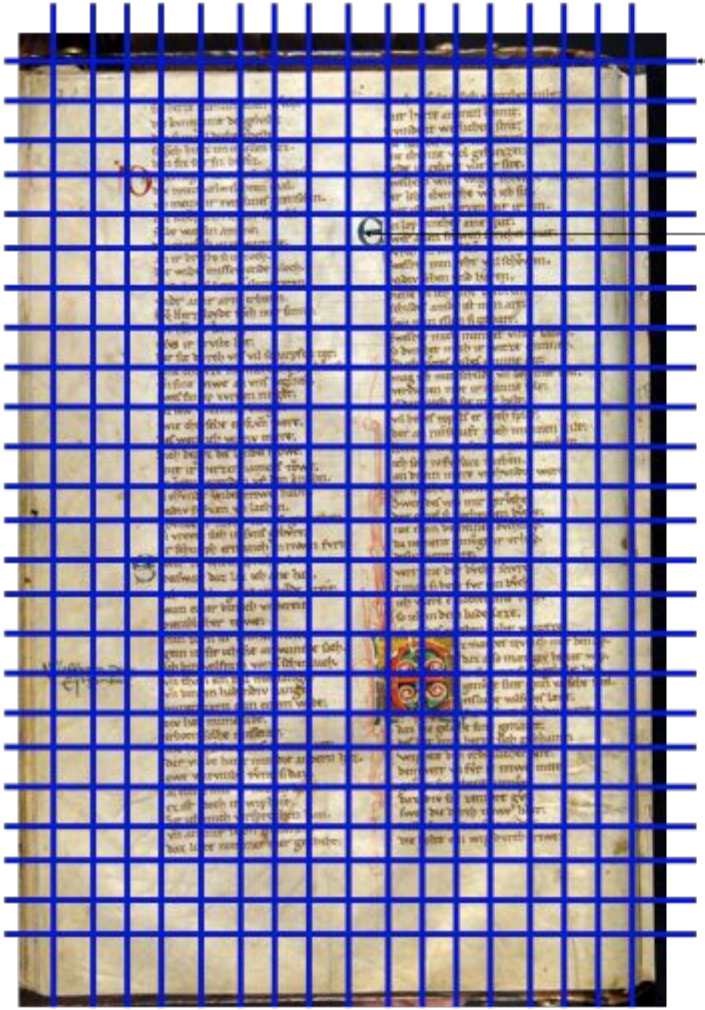
Introduction

- Challenge
  - Many variations: layout structure, decoration, degradation, writing style.



Why Machine Learning





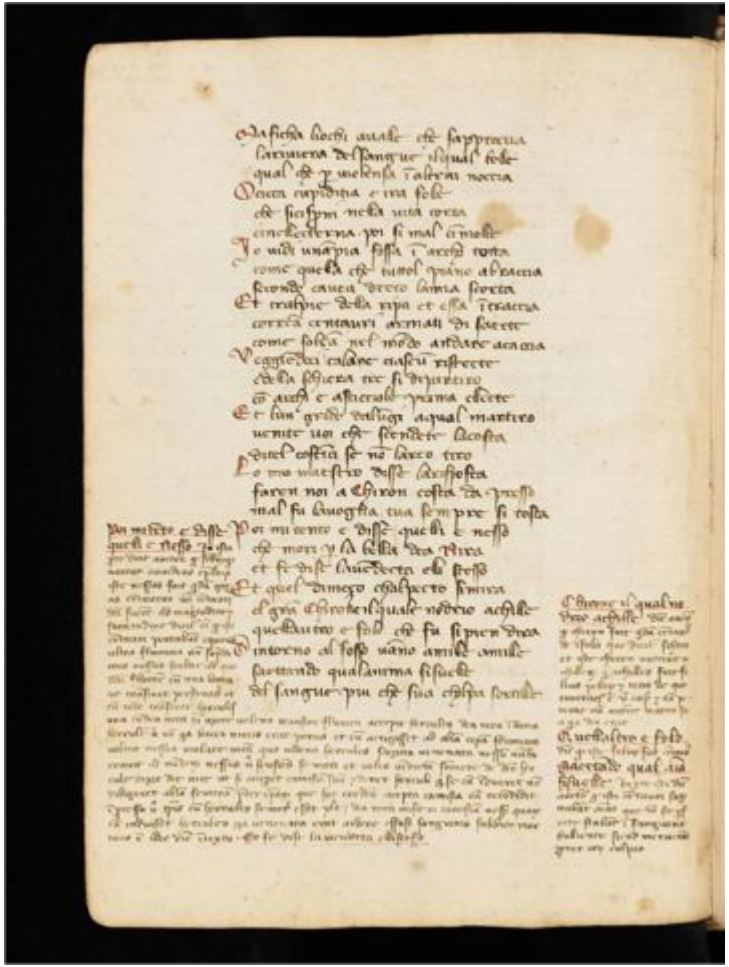
(Pre-processing)
Split a page into grids *x_i*.

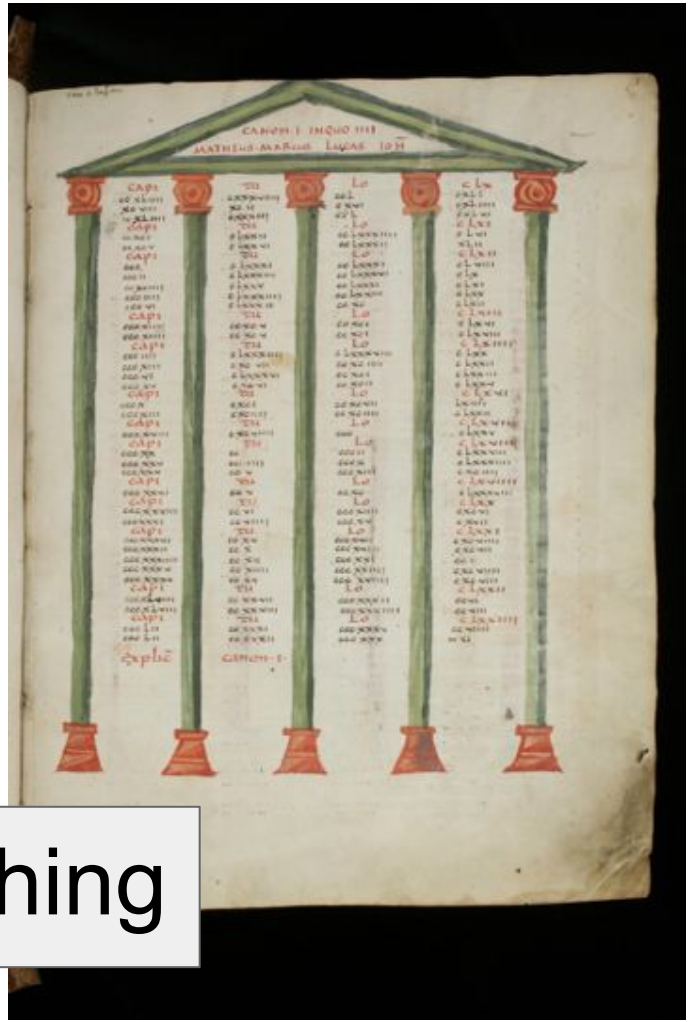
(Labeling)
for each *x_i*

if *rule_1*(*x_i*) **then**
  y_i ← TEXT // y_i is the label of x_i
else
  if *rule_2*(*x_i*) **then**
    y_i ← DECORATION
  ...

(Post-processing)
for each *x_i*

if *rule'_1*(*neighbors*(*x_i*)) **then**
  y_i ← TEXT
else
  if *rule'_2*(*neighbors*(*x_i*)) **then**
    y_i ← DECORATION
  ...





Dataset and Ground Truthing

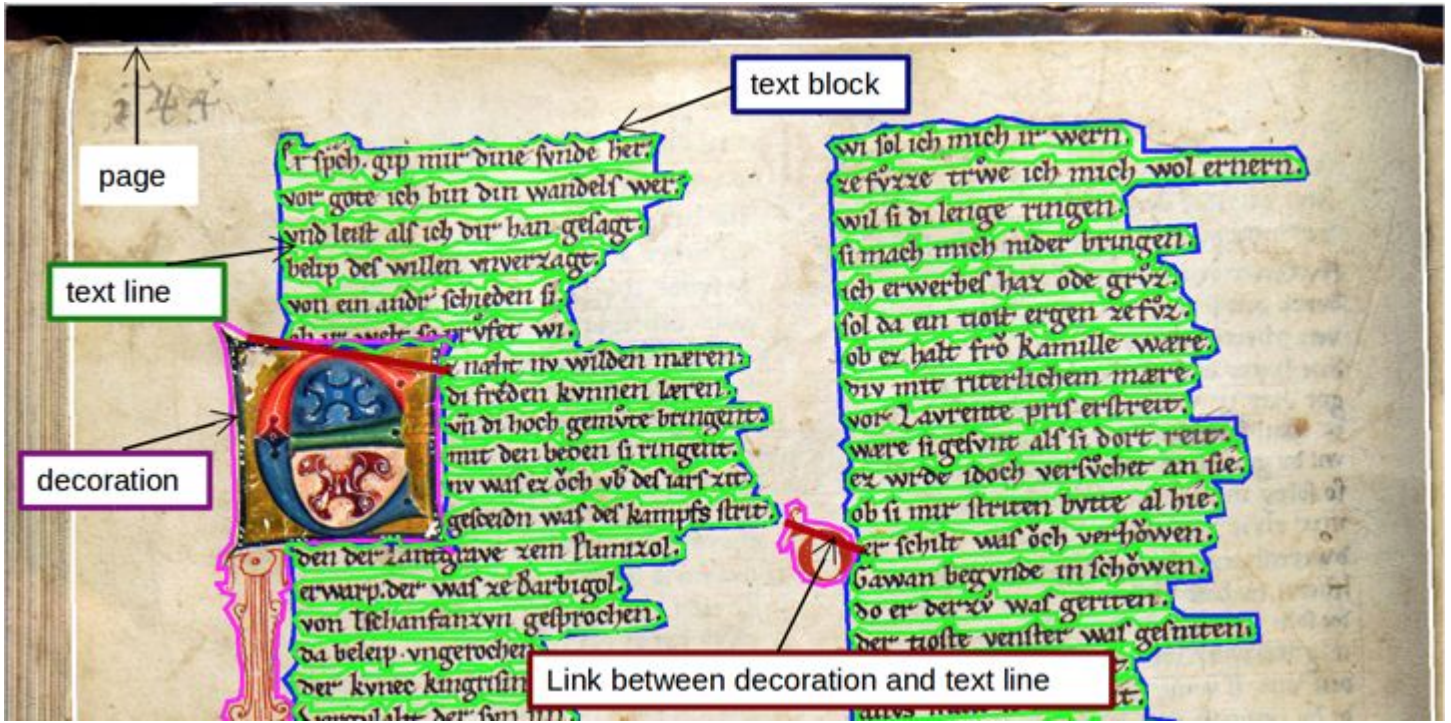
Dataset and Ground Truthing (Chen et al. DRR 15)

- Ground Truth
  - Used for training machine learning algorithms
  - Performance comparison
  - Problem: rarely publicly available for historical document images
- Objective
  - Ground truthing methodology (model and tool)
  - New datasets for layout analysis research

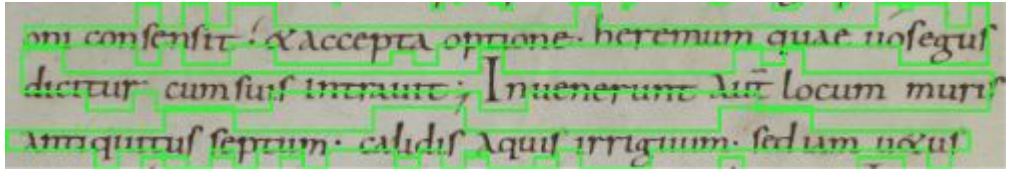
Chen et al., *Ground truth model, tool, and dataset for layout analysis of historical documents,* Proc. SPIE 9402, Document Recognition and Retrieval (DRR), 2015.

Dataset and Ground Truthing (Chen et al. DRR 15)

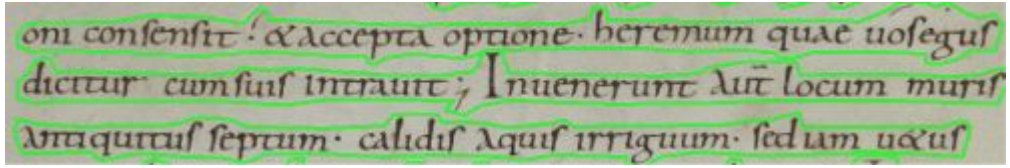
- Unconstrained Polygon



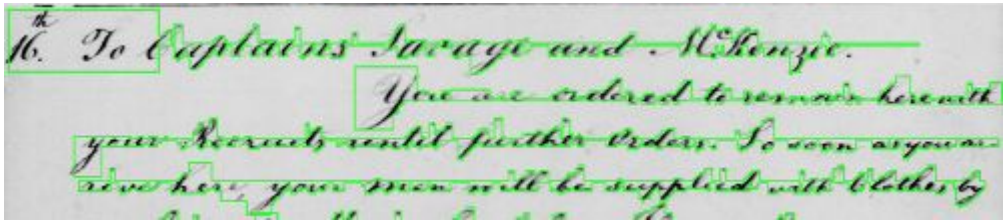
Previous



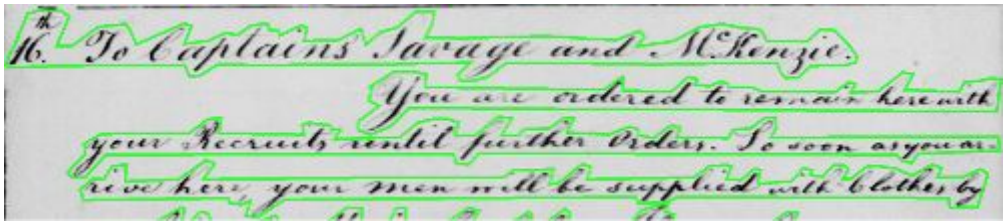
Current



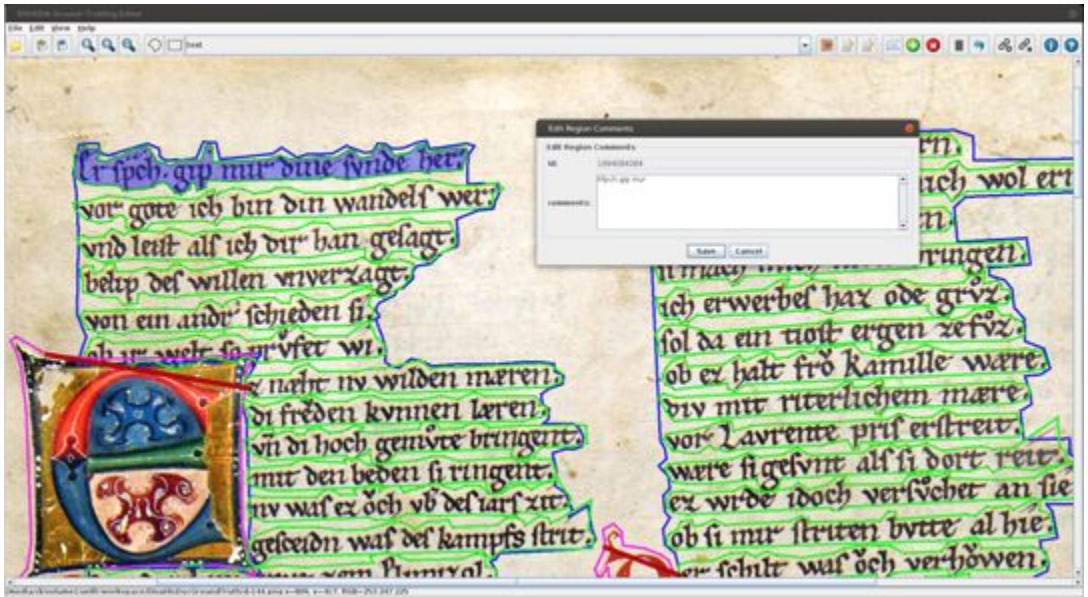
Previous



Current



- DIVADIA
  - Java based ground truthing tool for document image layout analysis
  - Target simplicity and efficiency of the ground truthing
  - 120 pages from three datasets have been annotated with DIVADIA.



DIVA (**D**ocument, **I**mage, and **V**oice **A**nalysis) research group ground-truthing tool for historical **D**ocument **I**mage **A**nalysis. *http://diuf.unifr.ch/main/hisdoc/divadia*

- DIVA-DB

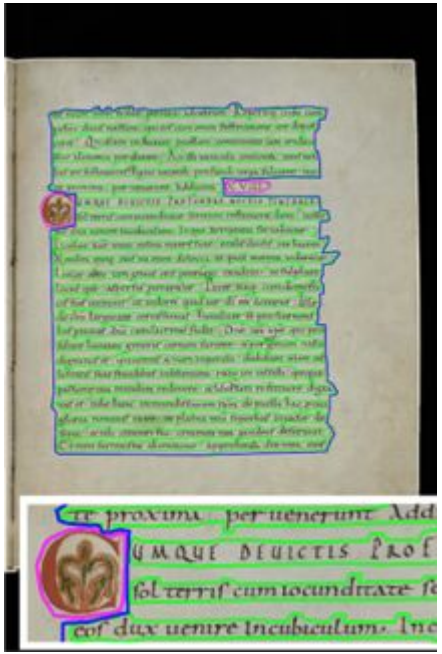
	GW	Parzival	St. Gall
Date (century)	18th	9th	13th
#Pages	20	30	61
Size (pixels)	2200 x 3400	2000 x 3008	1664 x 2496
Language	English	Old German	Latin
Writer	G. Washington et al.	Three writers	Single writer

http://diuf.unifr.ch/main/hisdoc/divadia
http://www.iam.unibe.ch/fki/databases/iam-historical-document-database

- Examples

George Washington
G. W. Papers, Library of Congress, (GW10)

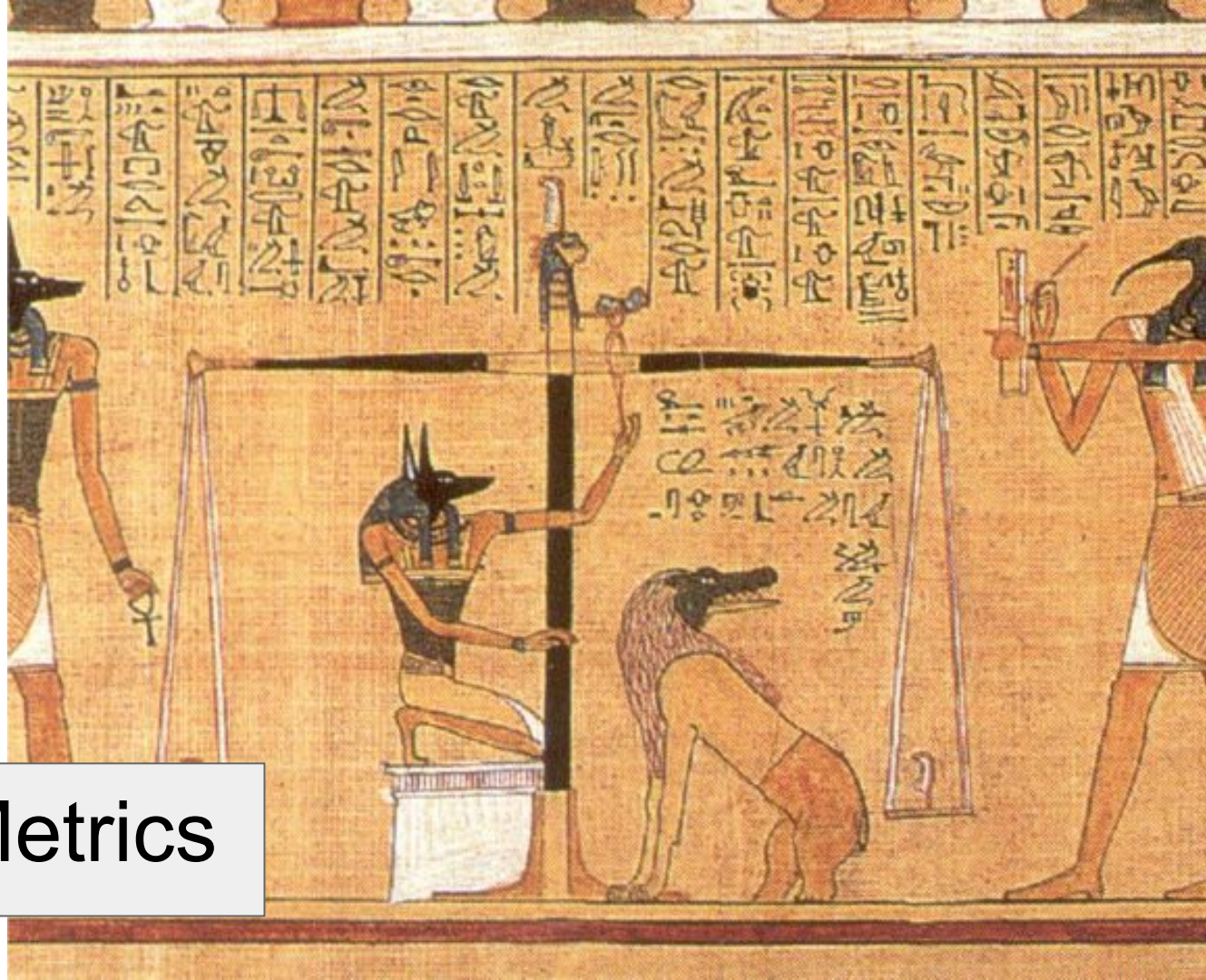
Parzival
Cod. 857, Abbey Library of St. Gall, (PAR23)



Saint Gall
Cod. Sang. 562, Abbey Library of St. Gall, (SG30)

Summary

- A new dataset for layout analysis research
- The dataset has been also used for the research of document degradation, document clustering, and several bachelor and master projects
- The ground-truth presentation is used in Handwritten Historical Document Layout Recognition Competition (ICDAR 17)



Evaluation Metrics

Evaluation Metrics

- Pixel accuracy

$$acc = \frac{\sum_i n_{ii}}{\sum_i \sum_j n_{ij}}$$

The number of pixels of class i predicted to belong to class j

- Mean accuracy

$$acc_{mean} = \frac{1}{|C|} \times \frac{\sum_i n_{ii}}{\sum_i \sum_j n_{ij}}$$

- Mean Intersection over Union (IU)

$$iu_{mean} = \frac{1}{|C|} \times \sum_i \frac{n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}$$

The total number of pixels in class i

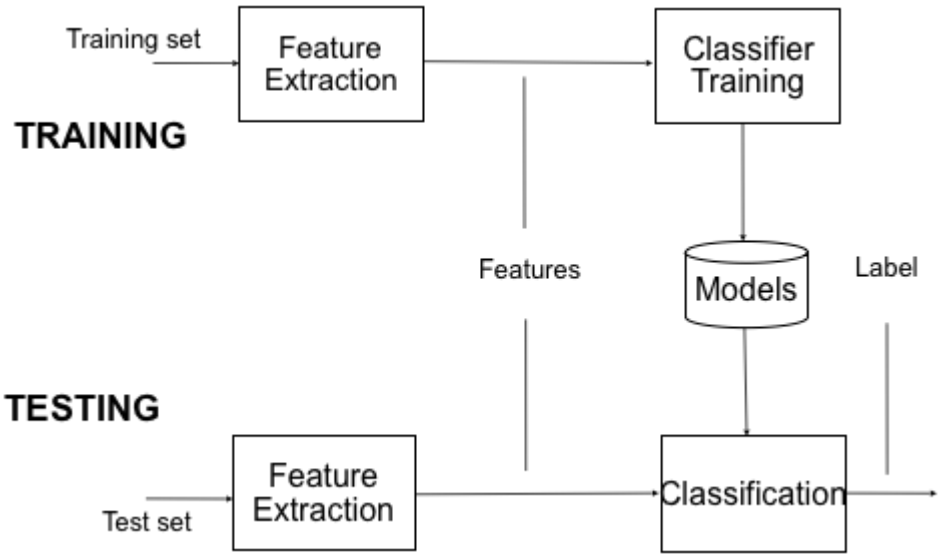
- Frequency Weighted Intersection over Union (f.w. IU)

$$iu_{weighted} = \frac{1}{\sum_k t_k} \times \sum_i \frac{t_i \times n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}}$$

Baseline Method

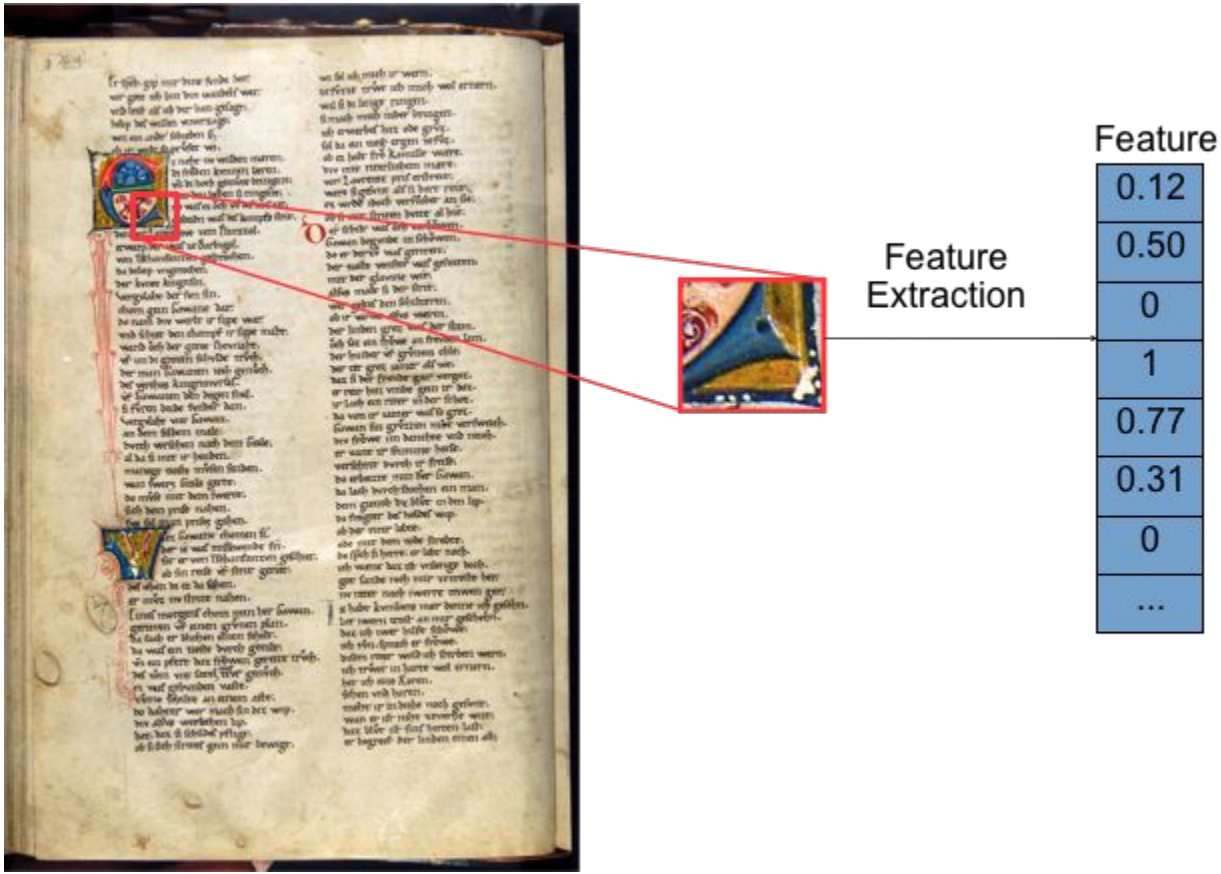
Baseline Method (Chen et al. ICFHR 14)

- Pixel labeling approach
  - **Training**: train a classifier with the class **label** and **features** (image representation, real-value vector) in a fixed window of each pixel
  - **Testing**: with the trained classifier, pixels are classified into classes



Chen et al., *Page Segmentation for Historical Document Images Using Color and Texture Features,* 14th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 488-493, 2014.

- Feature Extraction



Texture Features

- Gabor Filters
  - For a given pixel at position (x, y), get its **dominant orientation**

$$\theta = \arg\max I(x, y, \theta, \sigma)$$

$$I(x, y, \theta, \sigma) = \left| \sum_{x_1 = x - \frac{M}{2}}^{x + \frac{M}{2}} \sum_{y_1 = y - \frac{N}{2}}^{y + \frac{N}{2}} u(i, j) e^{\frac{-(x_1 - x)^2 - (y_1 - y)^2}{2\sigma^2}} e^{j\lambda(\cos\theta(x - x_1) + \sin\theta(y_1 - y))} \right|$$

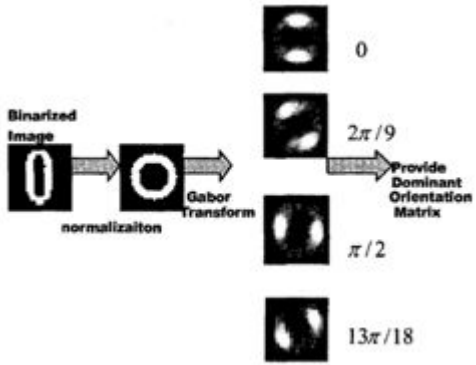


Figure taken from (Hu et al., 02)

Hu et al., *Recognition of gray character using Gabor filters*, ICIF, pp. 419-424, 2002.

Texture Features

- Local Binary Pattern (LBP)
  - LBP is based on **signs of differences** of a **circular neighboring pixels**

$$LBP_{P,R}(x,y) = \sum_{p=0}^{P-1} s(g_p - g_c)2^p$$

$$s(x) = \begin{cases} 1 & x \geq 1 \\ 0 & x < 0 \end{cases}$$

P : number of neighbors
R : radius
g_c : center pixel's grayscale value
g_p : neighbors grayscale value

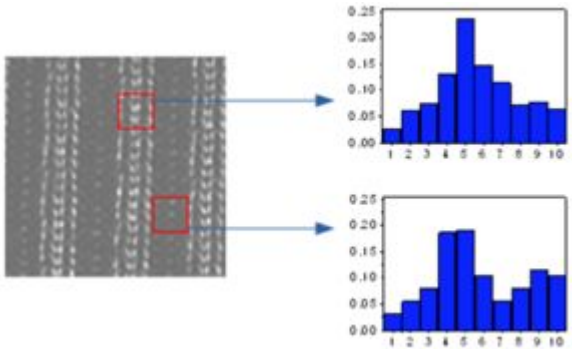


Figure taken from (Ahonen et al., 06)

Ahonen et al., *Face Description with Local Binary Patterns: Application to Face Recognition*, PAMI, vol. 28, issue. 12, pp. 2037-2041, 2006

Color and Location Features

- Color
  - For a given pixel, the color information (in the RGB color space) in its neighborhood is considered as features
    - Mean, Variance, Color smoothness, Horizontal mean, variance, and smoothness, Gradient
- Location
  - x, y coordinates of the given pixel

Experiments

best result

	G. Washington				*Parzival*				*St.Gall*			
	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU
co	73	79	60	62	**84**	60	**51**	**72**	89	84	75	82
co+lo	71	78	57	60	82	62	50	70	90	85	76	82
te	85	**85**	**67**	77	72	**69**	42	63	**92**	78	65	85
te+lo	**88**	79	**67**	**79**	83	61	48	71	**92**	78	65	85
co+te	72	72	56	63	81	64	49	70	81	59	49	70
co+te+lo	68	78	57	58	50	61	50	71	**92**	**93**	**87**	**85**
(Baechler and Ingold, 2011)	77	79	63	68	82	58	49	69	61	55	45	48
	CB55				*CSG18*				*CSG863*			
	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU
co	78	48	35	70	82	25	21	68	80	48	35	71
co+lo	**81**	54	**39**	**72**	36	39	16	26	**83**	46	**36**	73
te	69	48	30	60	66	48	27	58	72	45	30	62
te+lo	61	38	22	54	65	**49**	26	57	70	**50**	31	61
co+te	78	51	35	70	**83**	25	21	70	81	49	**36**	72
co+te+lo	80	49	35	71	**83**	25	21	70	**83**	44	**36**	**74**
(Baechler and Ingold, 2011)	79	**58**	**39**	71	78	47	**32**	**71**	74	43	29	64

- lo: location, co: color, te: texture
- window size is tuned on the validation set

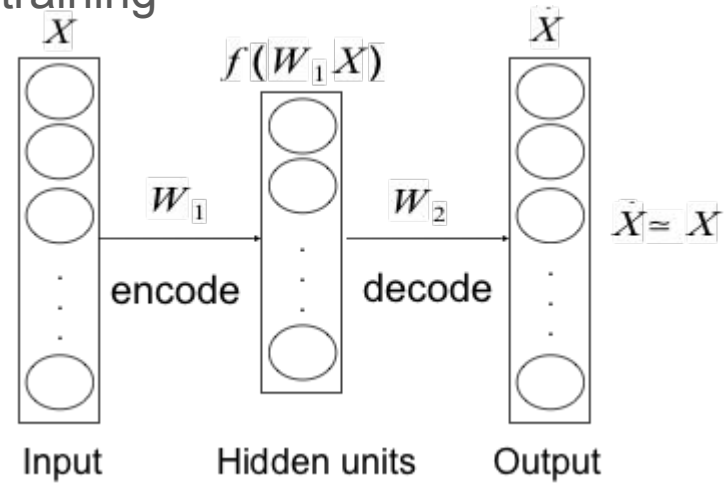
Summary

- Baseline method for page segmentation
- Color and texture features for the segmentation task
- **Good hand-crafted features are dataset dependent**

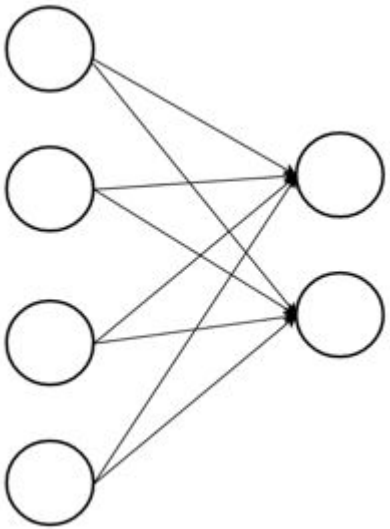

Feature Learning

Feature Learning

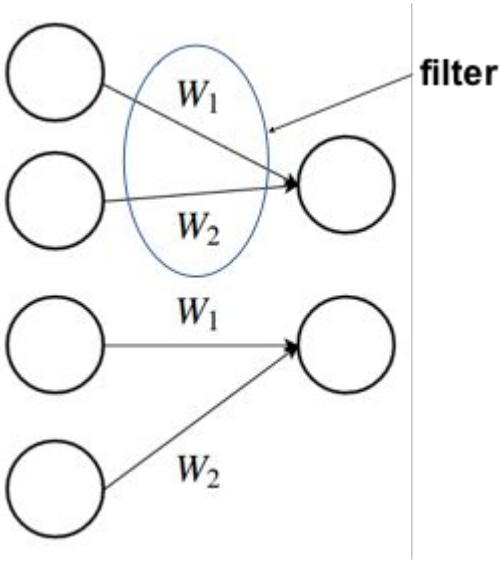
- Autoencoder
  - A feed-forward neural network aims to learn a compressed representation (**encoding**) of data
  - The network is trained to “recreate” (**decode**) the input, i.e., the input and the target data are same/similar
  - The output (activations) of the hidden units can be used as features for supervised training



- Convolution
  - For a given image of size 100x100 pixels. We want to extract 100 features with autoencoder. Then we have 100x100x100=**10^6** parameters to learn
  - Images have the property of being stationary, i.e., one part of the image is the same as any other parts



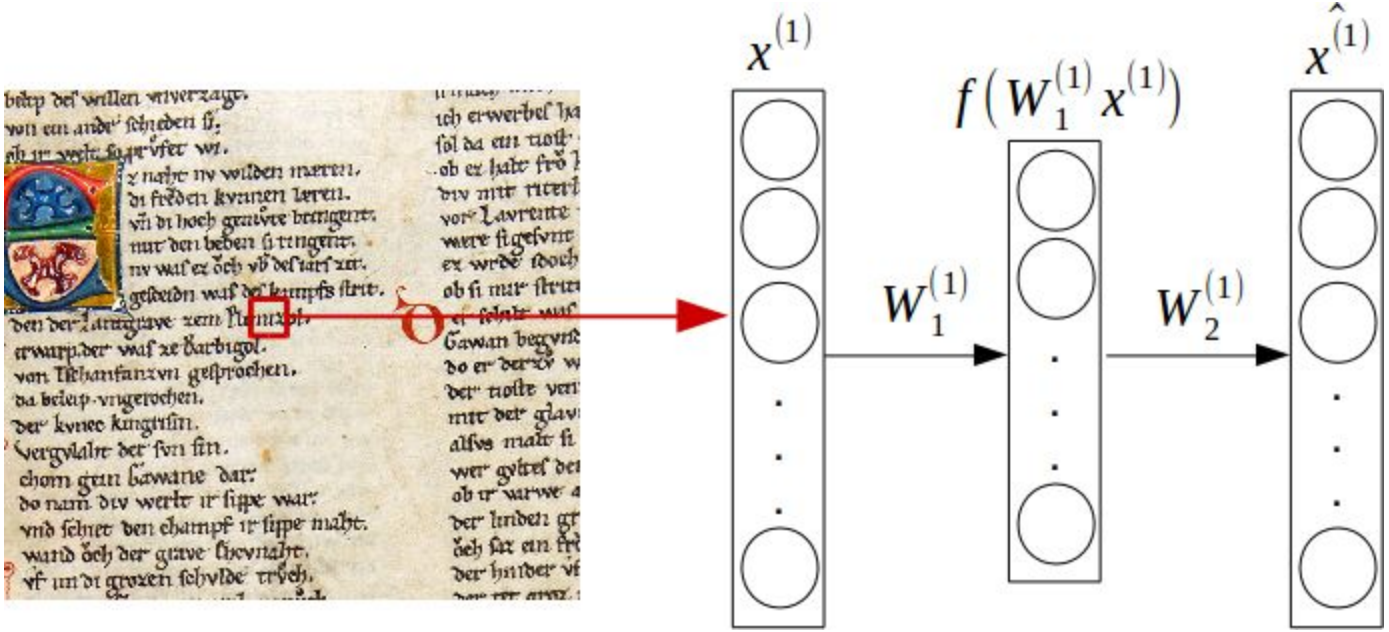
Fully Connected Layer



Convolution Layer

Feature Learning (Chen et al. ICDAR 2015)

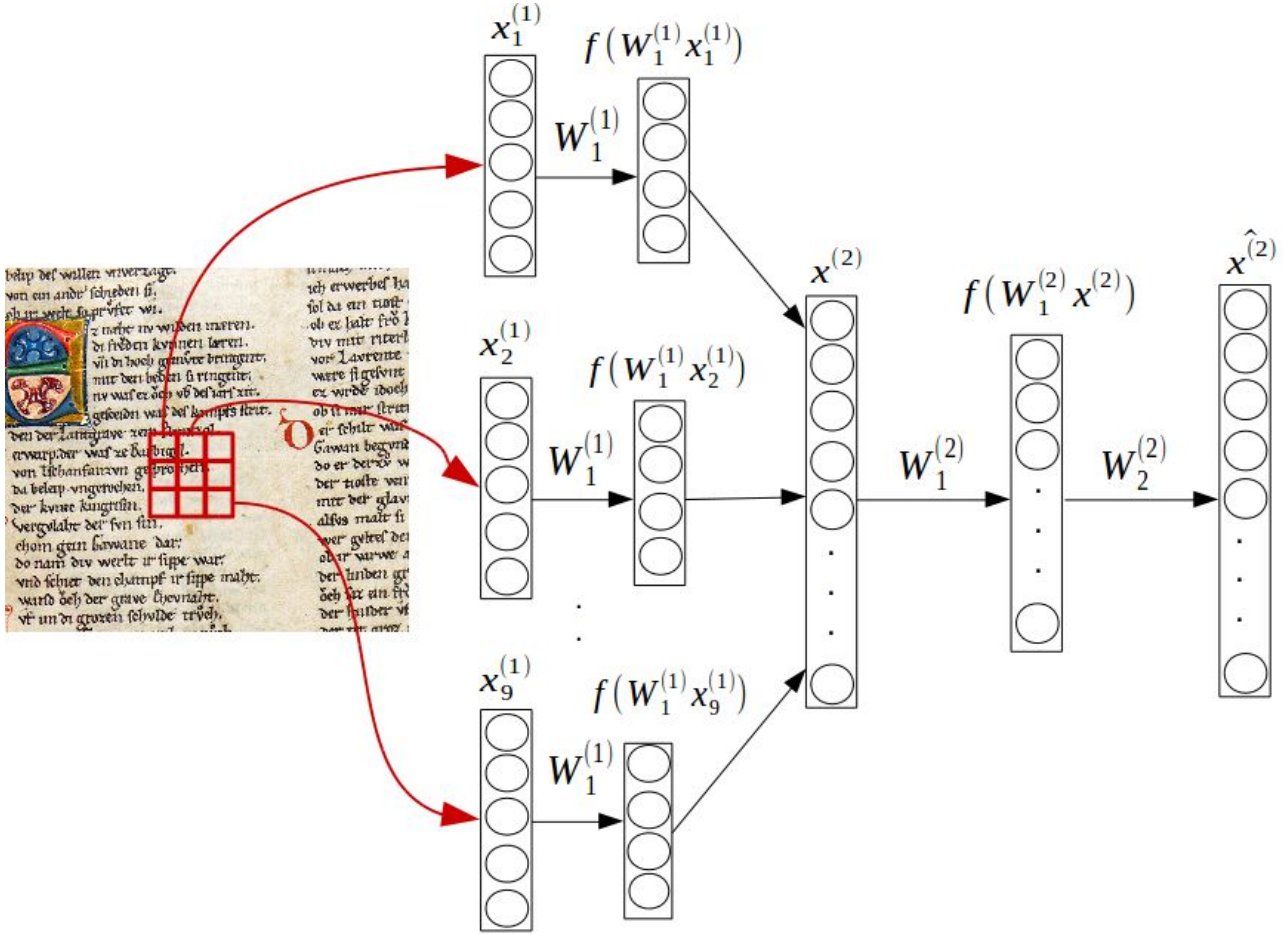
- Level 1



- patch size = 5x5, #hidden units = 40, #patches = 500K

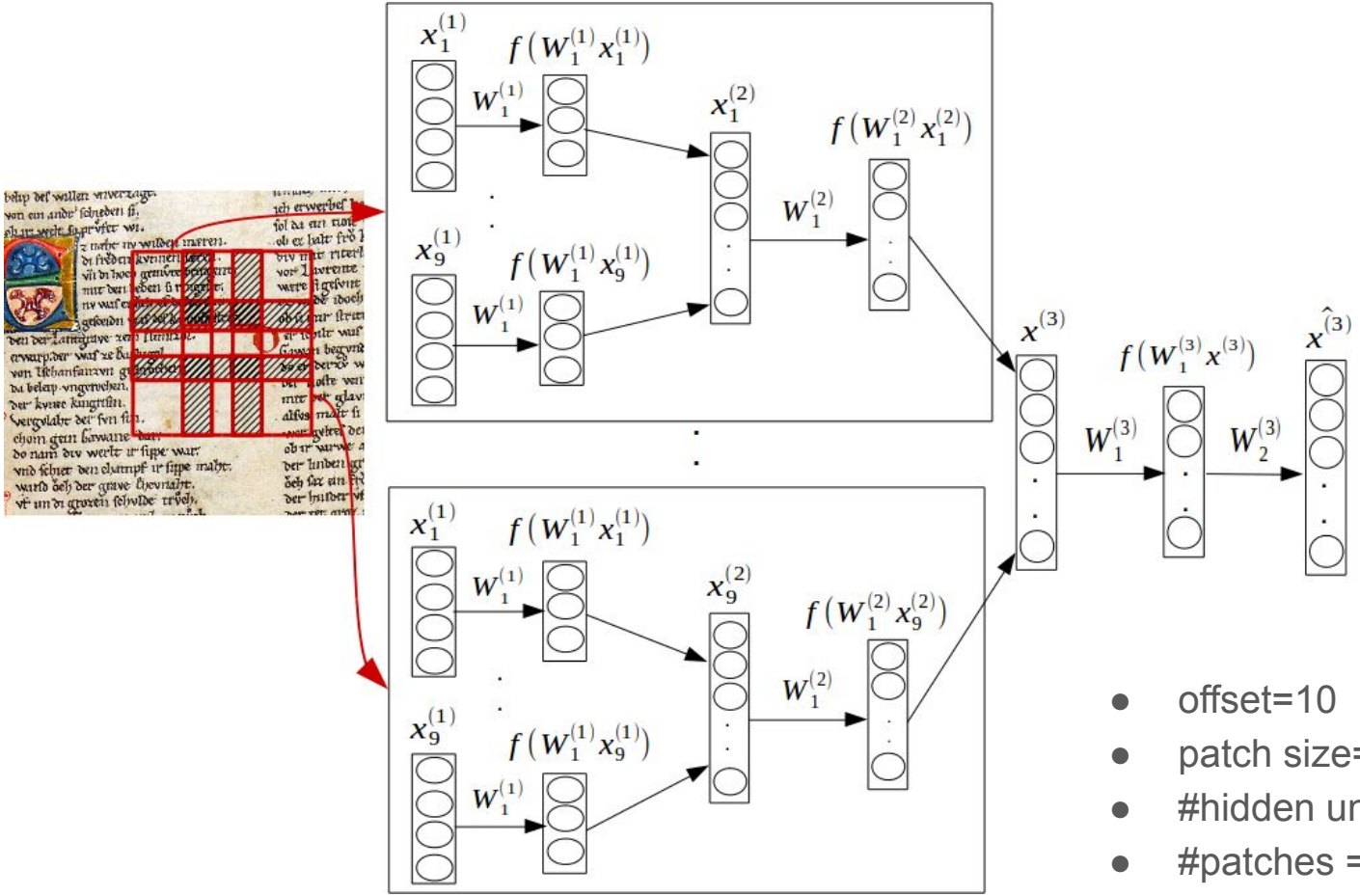
Chen et al., *Page segmentation of historical document images with convolutional autoencoders*, ICDAR, pp. 1011-1015, 2015.

- Level 2



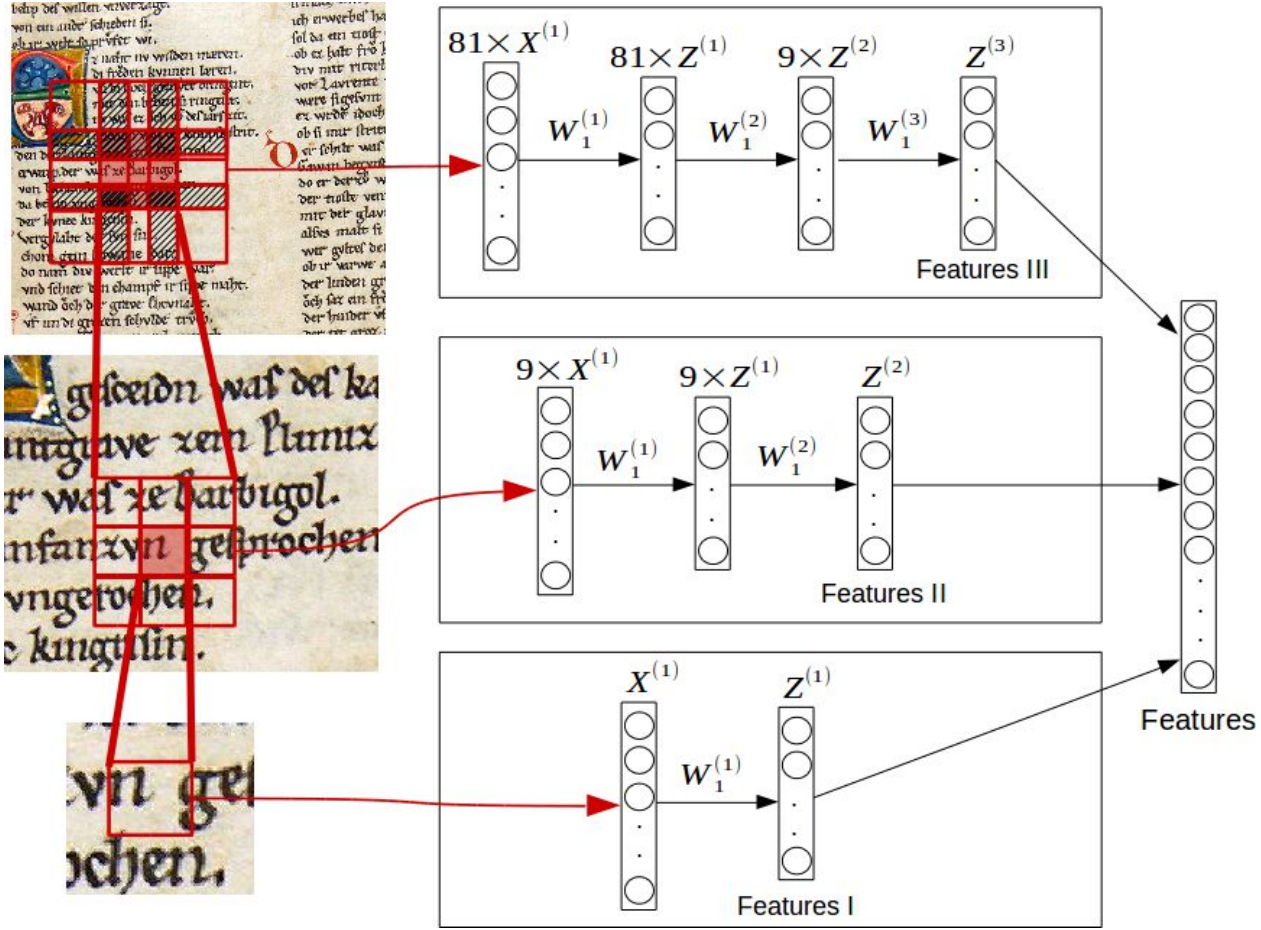
- patch size = (3x5)^2
- #hidden units = 30
- #patches = 500K

- Level 3



- offset=10
- patch size=(3x5+2xoffset)² = 35²
- #hidden units = 30
- #patches = 200K

- Feature Extraction



Experiments

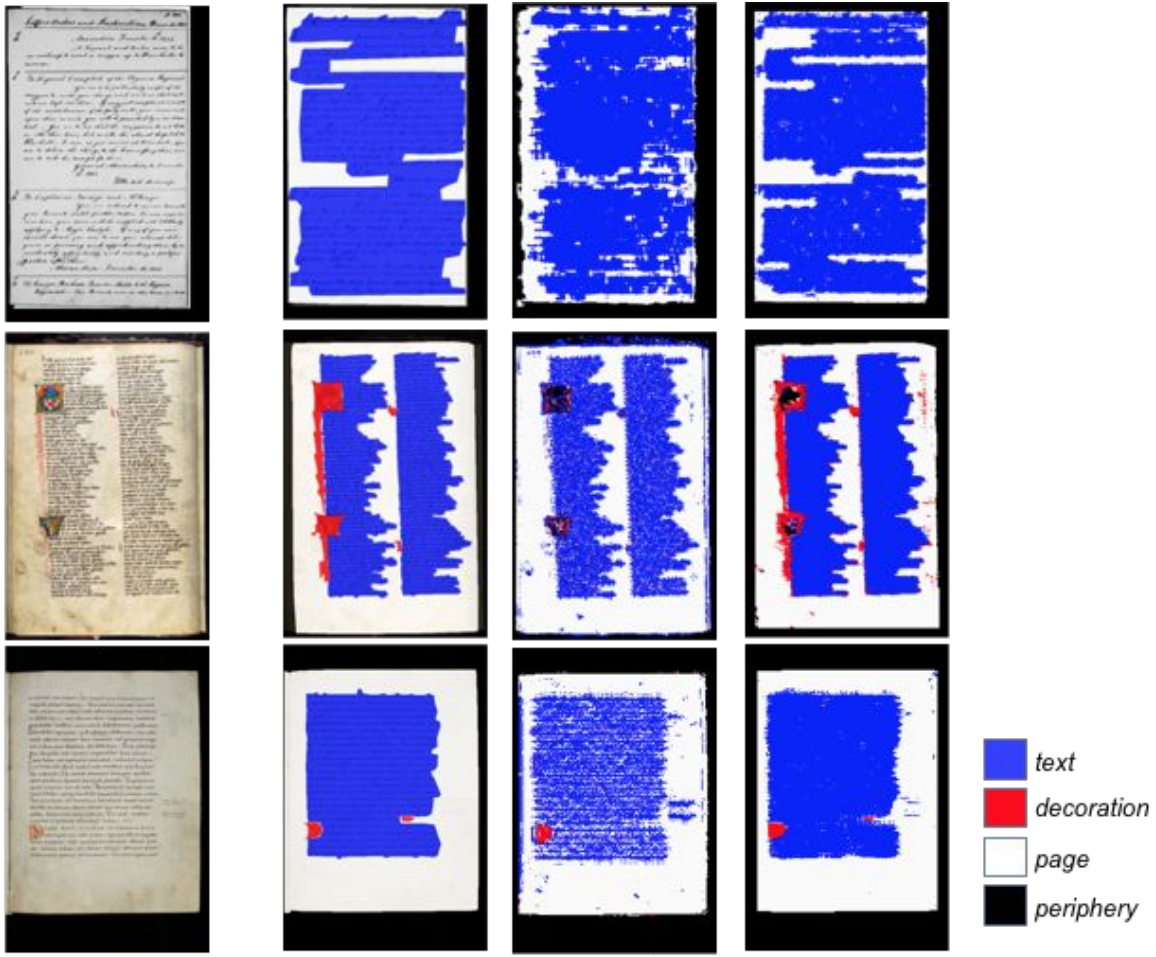
	G. Washington				*Parzival*				*St.Gall*			
	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU
hand-crafted features	82	84	64	73	89	64	57	81	94	87	78	91
learned features	**87**	**91**	**74**	**81**	**93**	**75**	**65**	**89**	**96**	**93**	**87**	**94**
	CB55				*CSG18*				*CSG863*			
	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU
hand-crafted features	84	62	45	76	**86**	**65**	**49**	**79**	**84**	50	38	76
learned features	**87**	**76**	**52**	**79**	85	59	41	77	83	**65**	**44**	**77**

comparable improved degraded

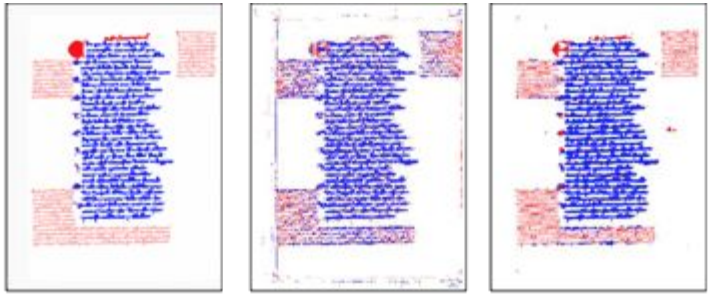
GW

Parzival

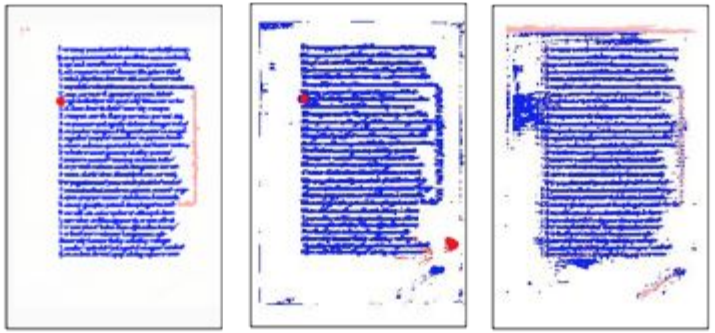
St. Gall



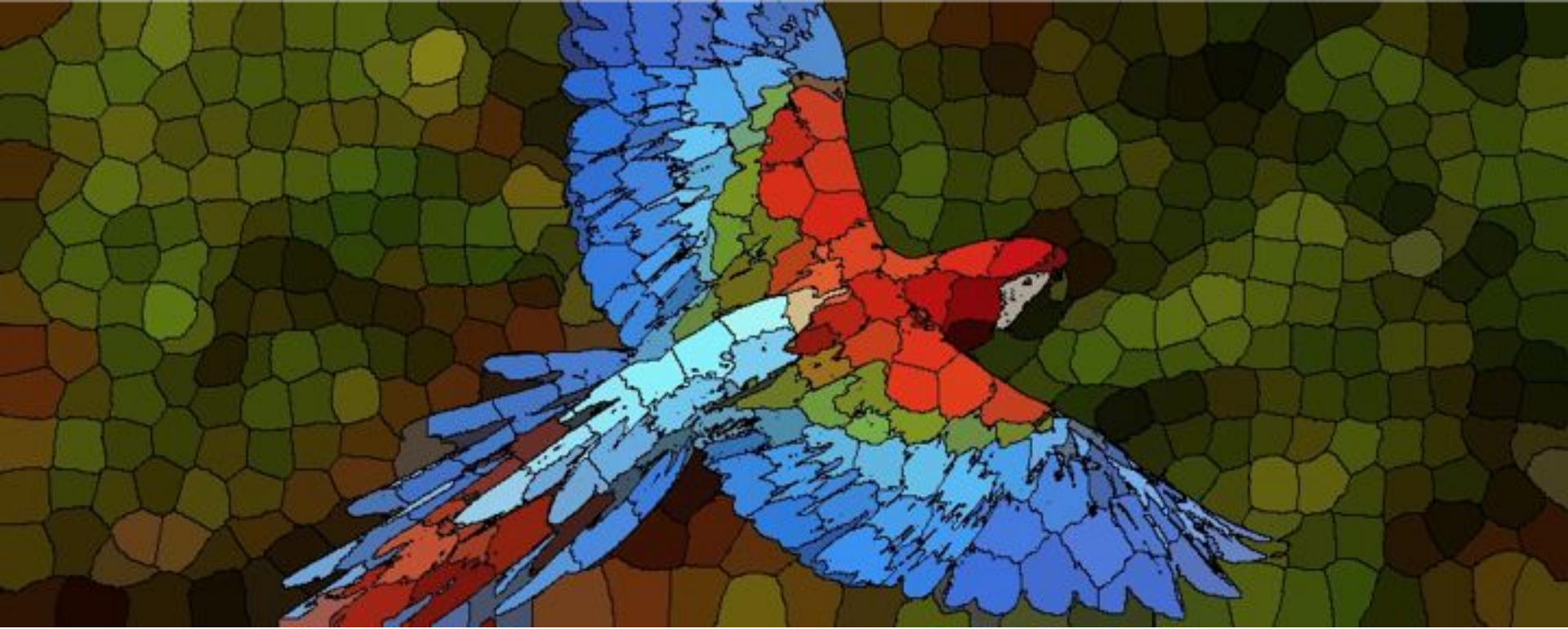
CB55



CSG18



CSG863



Superpixels

The work is under the supervision of Prof. Cheng-Lin Liu and supported by the visiting student program of China Academy of Sciences from June to September 2015.

Superpixels for Page Segmentation (Chen et al. DAS 16)

- Motivation
  - Speed up the segmentation
- Superpixels
  - Image patches which contain pixels belong to same object
  - Instead of labeling all pixels, **labeling the center pixel** of each superpixel and the rest pixels are assigned with the same label
- Criteria for superpixel algorithms evaluation
  - **Increase** speed and **improve** (or not degrade) the quality of segmentation

Chen et al., *Page Segmentation for Historical Document Images Based on Superpixel Classification with Unsupervised Feature Learning*, 12th IAPR International Workshop on Document Analysis System (DAS), pp. 299-304, 2016.

Superpixel algorithms

- Watershed approach (WS) (Vincent and Sollie 1991)
  - Image is viewed as a topological map where **pixel intensity is represented by its gradient**
  - **O(N logN)** : N is the number of pixels
- Mean shift (MS) (Comanicu and Meer 2002)
  - The **means** of the data samples within each window are computed
  - The windows are **shifted** to the locations equal to their previously computed means
  - **O(N^2)**
- Felzenszwalb and Huttenlocher (FH) (Felzenszwalb and Huttenlocher 2004)
  - Each superpixel is represented by a **minimum spanning tree**
  - **O(N logN)**
- Simple linear iterative clustering (SLIC) (Achanta et al. 2012)
  - **K-means clustering** in the combined five dimensional color and coordinate space
  - **O(k N I)** : k is the number of superpixels. I is the number of iterations.

Experiments

	G. Washington				*Parzival*				*St.Gall*			
	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU
pixel	87	**91**	**74**	81	**93**	75	**65**	**89**	**96**	**93**	**87**	**94**
superpixel	**88**	90	**74**	**82**	92	**77**	62	86	**96**	91	86	93
	CB55				*CSG18*				*CSG863*			
	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU
pixel	87	**76**	**52**	79	85	**59**	**41**	**77**	83	**65**	**44**	**77**
superpixel	**88**	69	50	**80**	**86**	51	40	76	**87**	51	43	**77**

comparable improved degraded

Run time in seconds

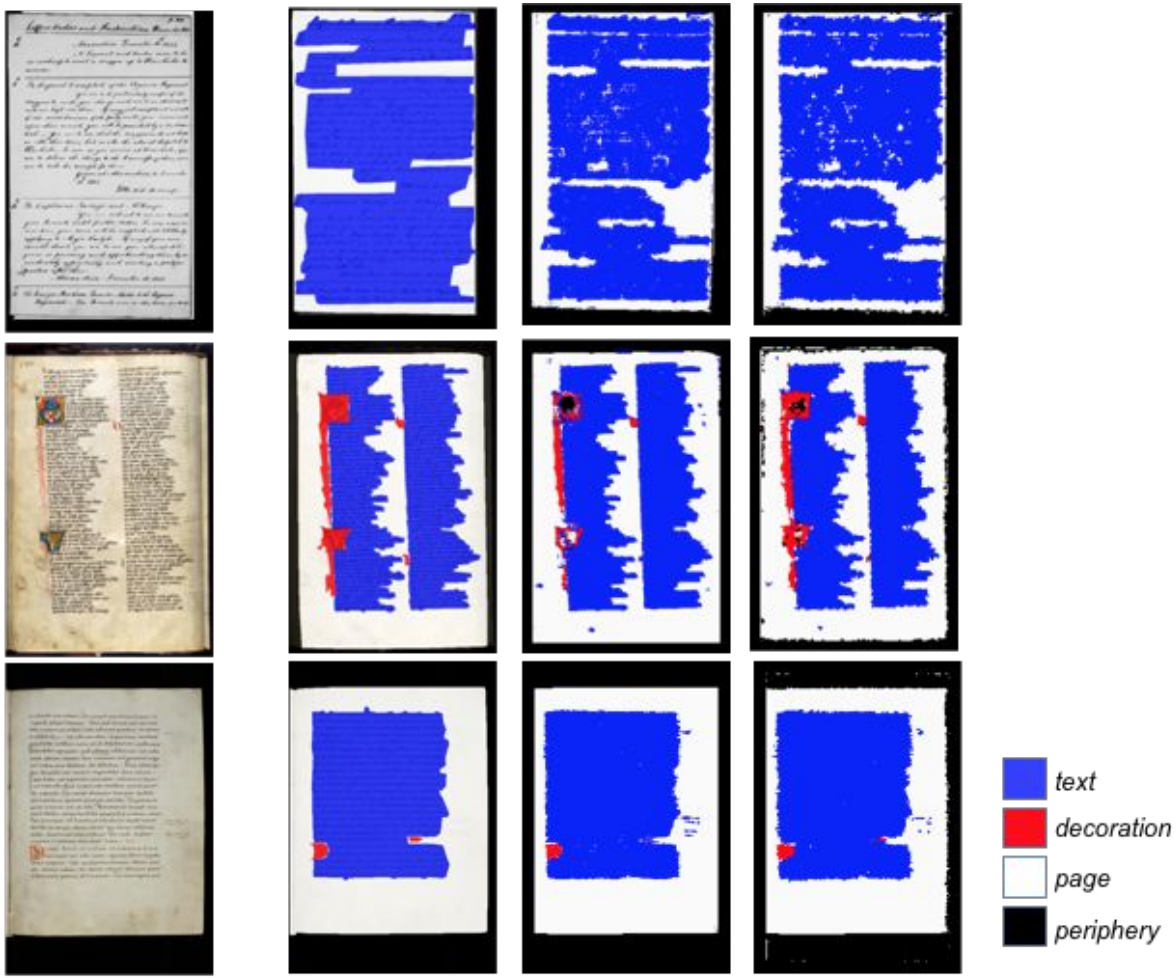
	G. Washington	*Parzival*	*St.Gall*	*CB55*	*CSG18*	*CSG863*
pixel	30	22	16	31	18	16
superpixel	**8**	**6**	**5**	**8**	**7**	**5**

- Environment: Intel Core i7-3770 3.4 GHz, 16GB RAM
- Programming Language: Java
- Operating System: Ubuntu Linux 14.04 LTS
- Superpixels are generated by using SLIC
- Number of superpixels per image: **20K**

GW

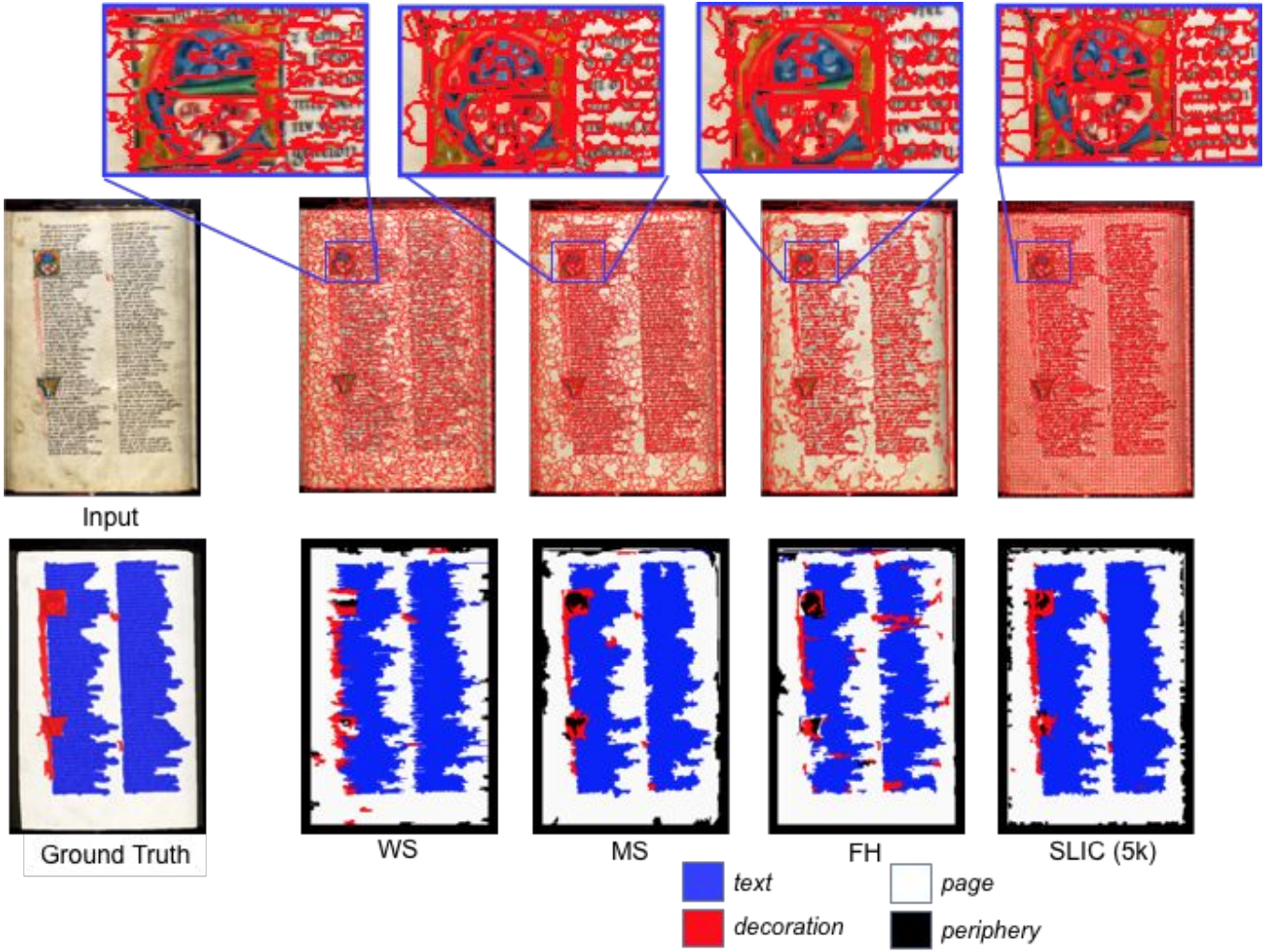
Parzival

St. Gall



Input | Ground Truth | Pixel Labeling | Superpixel Labeling


CB55
CSG18
CSG863
Input
Ground Truth
Pixel Labeling
Superpixel Labeling
text
decoration
page
comment


Input
Ground Truth
WS
MS
FH
SLIC (5k)
text
decoration
page
periphery

Structure Learning

The work is under the supervision of Prof. Cheng-Lin Liu and supported by the visiting student program of China Academy of Sciences from June to September 2015.

Structure Learning

- “Normal” Machine Learning
  - Inputs X can be any kind of objects (images, text, audio, …)
  - Output is a **real number** (label, score, …)

$$f : X \Rightarrow R$$

- Structured Output Learning
  - Inputs X can be any kind of objects (images, text, audio, …)
  - Output Y is a **complex (structured) object** (image, layout, labels, …)

$$f : X \Rightarrow Y$$

CVPR 2011 Tutorial, Structured Prediction and Learning in Computer Vision.
Nowozin and Lampert, *Structure Learning and Prediction in Computer Vision*, Foundations and Trends in Computer Graphics and Vision, pp. 185-365, 2011.

Structure Learning

- Limitations of our previous method
  - **Context information** is not taken into account
  - Pixels are labeled **independently**, i.e., for a given pixel i

$$\hat{y}_i = \arg\max_{y_i \in L} P(y_i|x_i)$$

predicted label — label set — local features

- Structured Learning method
  - **Combine** the local and context information
  - Predict the labels of pixels **jointly**, i.e.,

$$\hat{Y} = \arg\max_{Y \in L} P(Y|X)$$

$$Y = \{y_0, y_1, \cdots, y_n\} \quad X = \{x_0, x_1, \cdots, x_n\}$$

Conditional Random Field (CRF)

- Objective

$$\hat{Y} = \arg\max_{Y \in L} P(Y|X)$$

- Energy Function

patch index

$$\psi(X, Y, \lambda) = \sum_j \sum_i \lambda_j f_j(X, Y, i)$$

weights

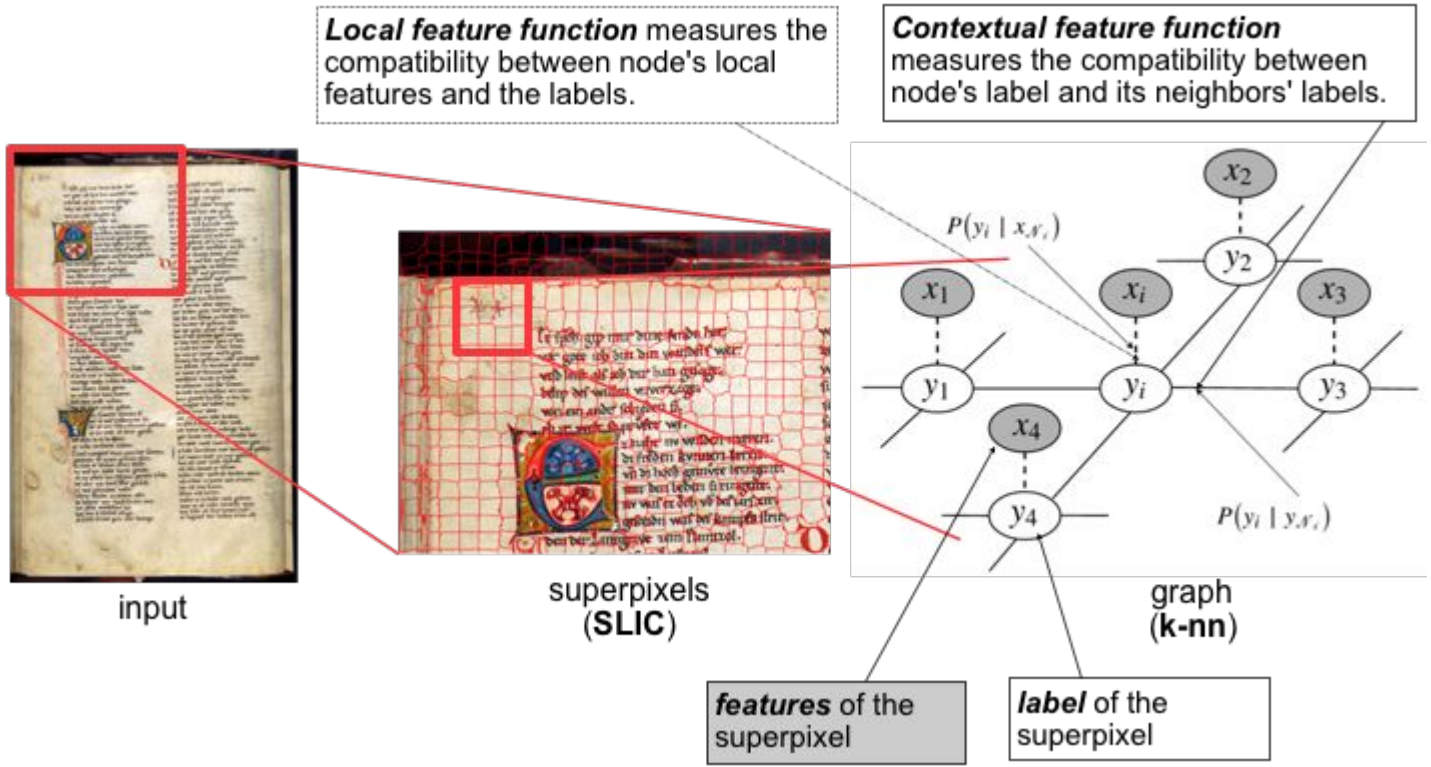
feature function

- Conditional Probability

$$P(Y|X;\lambda) = \frac{1}{Z(X,\lambda)} \exp(\psi(X, Y, \lambda))$$

$$Z(X,\lambda) = \sum_{\hat{Y}} \exp(\psi(X, \hat{Y}, \lambda)) \quad \longleftarrow \text{normalization function}$$

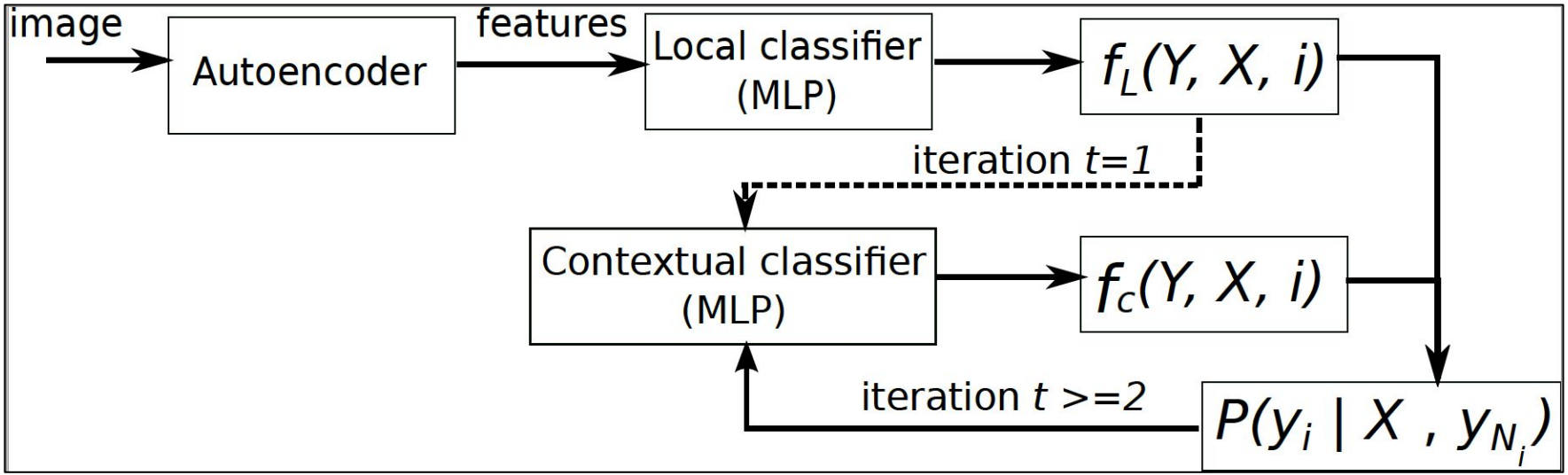
Page Segmentation with Structure Learning
(Chen et al. ICFHR 16)



Chen et al.,*Page Segmentation for Historical Handwritten Document Images Using Conditional Random Fields*, 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 90-95, 2016.

Inference (predict labels)

- Iterated Conditional Modes (ICM)
  - Avoid computing the normalization factor *Z*
  - Initialize labels with local classifier
  - **Iteratively** label each node with the combination of local and contextual classifiers, s.t. $y_i = \arg\max_{y_i \in L} P(y_i | y_{N_i}, X)$



Parameter Learning (estimate weights)

- Pseudo Likelihood (PL)
  - Local approximation of training data
  - Avoid computing normalization factor *Z* over whole image

$$\hat{\lambda} = \arg\max_{\lambda} \prod_{i=1}^{n} P(y_i | y_{N_i}, x_i; \lambda)$$

fixed with ground-truth labels

- Take log of the PL

$$J(\lambda) = \sum_{i=1}^{n} (\psi - \log(\sum_{y'} \exp(\psi)))$$

Parameter Learning (estimate weights)

- Stochastic gradient ascent

$$\lambda_j = \lambda_j + \eta\left(\sum_{i=1}^{n} f_j(y_i, x_i) - \sum_{y'} P(y_i' \mid y_{N_i}, x_i; \lambda) f_j(y_i', x_i)\right)$$

Maximize what we observe

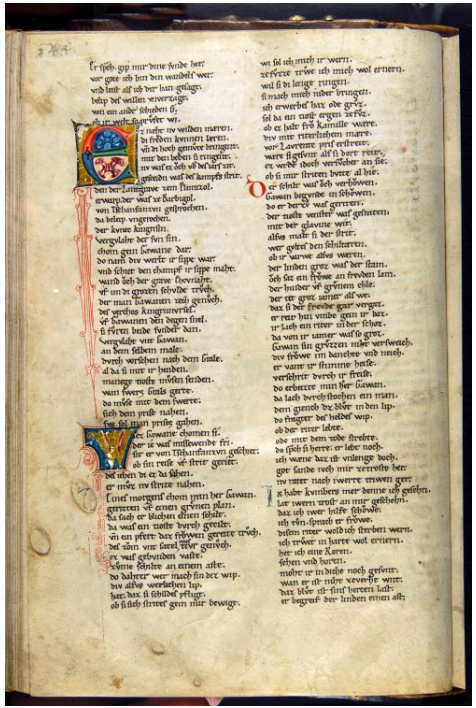
Minimize what we don't observe

Output of the feature function under the true label

Expected output of the feature function under other labels

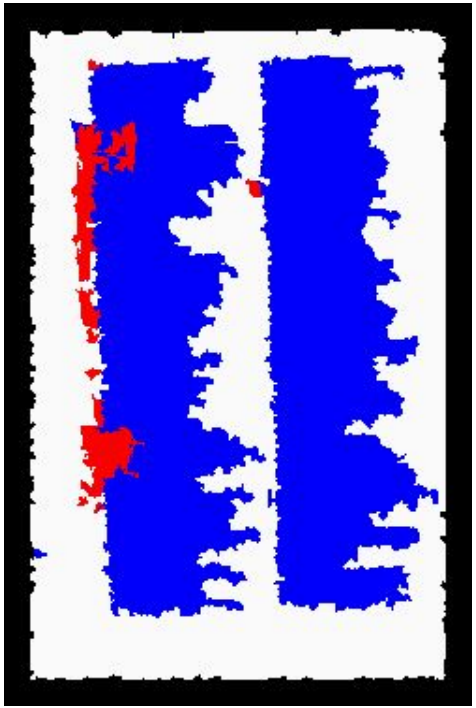
CNN for Page Segmentation of Historical Document

(Chen et al., ICDAR 2017)



Input

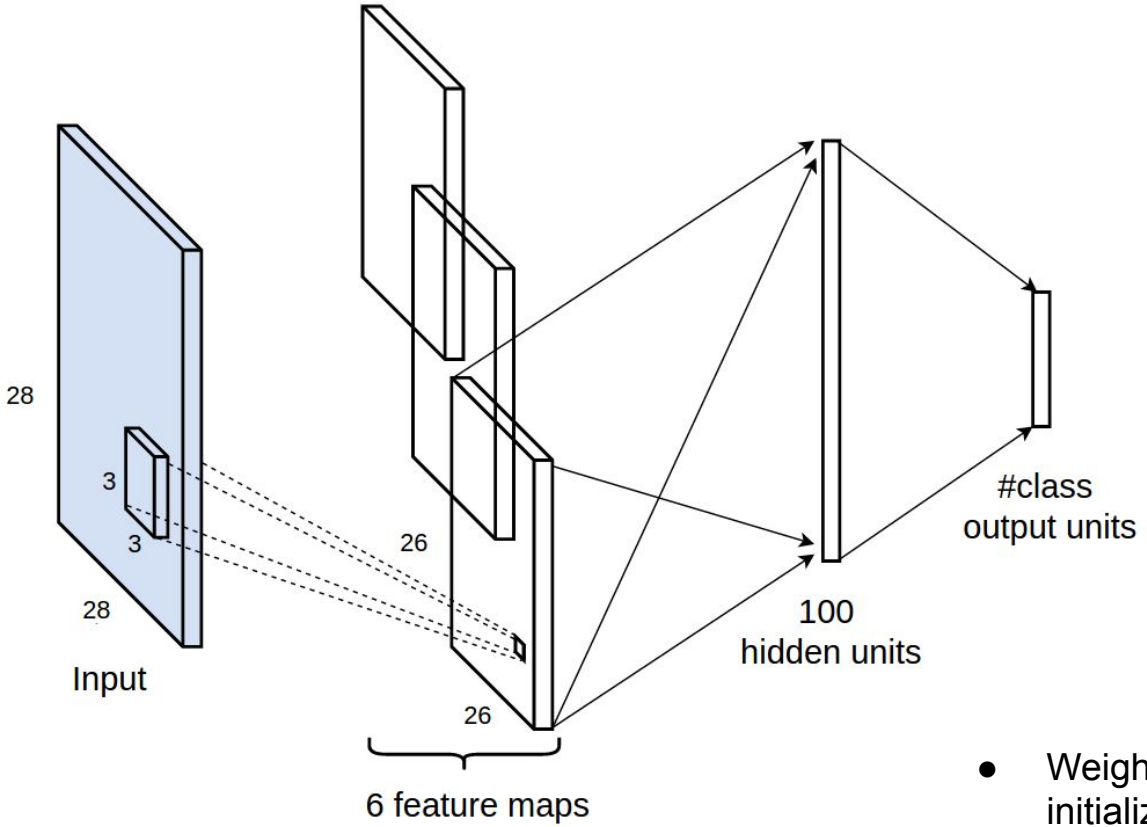
Superpixels (SLIC)



Labeling(CNN)

Chen et al.,*Convolutional Neural Networks for Page segmentation of Historical Document Images*, 14th International Conference on Document Analysis and Recognition (ICDAR), pp 965-970, 2017.

CNN Structure

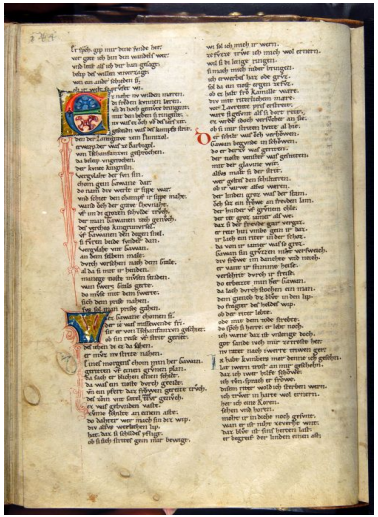


- Weights: Xavier initialization
- Stride size: 1

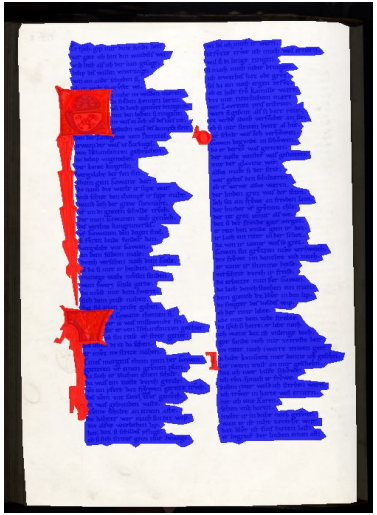
Experiments

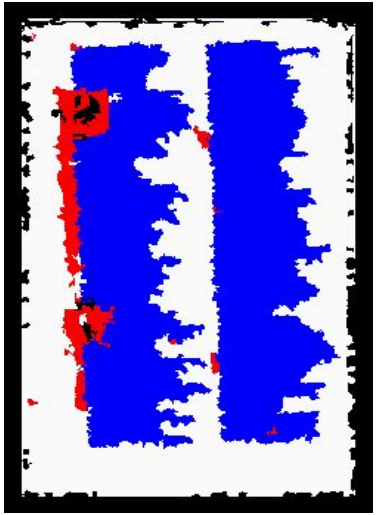
Performance (in percentage) of superpixel labeling with only local MLP, CRF, and the proposed CNN.

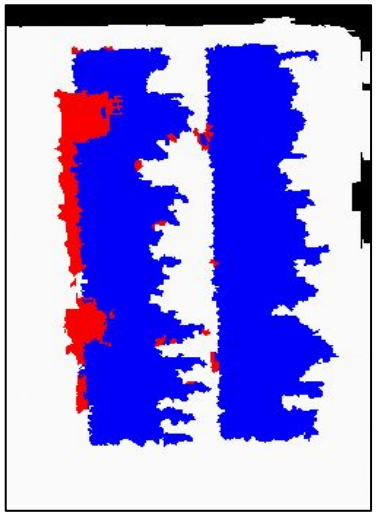
	G. Washington				*Parzival*				*St.Gall*			
	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU
Local MLP	87	89	75	83	91	64	58	86	95	89	84	92
CRF	**91**	90	76	85	93	70	63	88	97	88	84	94
CNN	**91**	**91**	**77**	**86**	**94**	**75**	**68**	**89**	**98**	**90**	**87**	**96**
CNN (max pooling)	**91**	90	**77**	**86**	**94**	**75**	**68**	**89**	**98**	**90**	**87**	**96**
	CB55				*CSG18*				*CSG863*			
	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU	pixel acc.	mean acc.	mean IU	f.w. IU
Local MLP	83	53	42	72	83	49	39	73	84	54	42	74
CRF	84	53	42	75	86	47	37	77	86	51	42	78
CNN	**86**	59	47	**77**	**87**	**53**	41	79	**87**	**58**	**45**	**79**
CNN (max pooling)	**86**	**60**	**48**	**77**	**87**	**53**	**42**	**80**	**87**	57	**45**	**79**

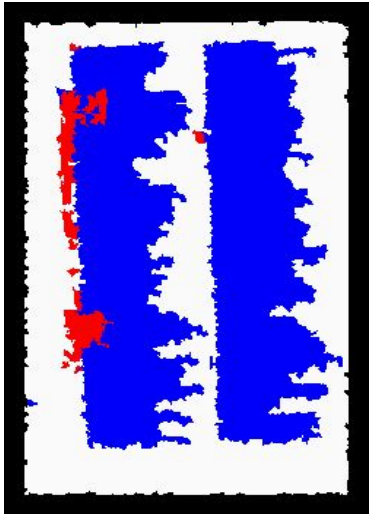


Input

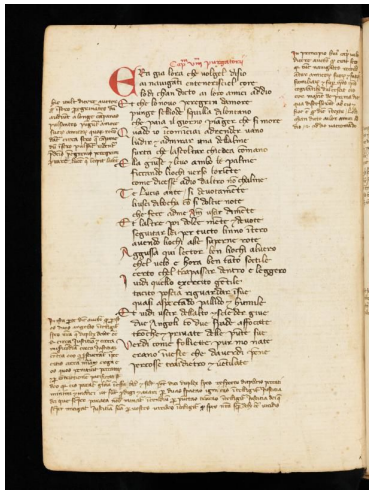




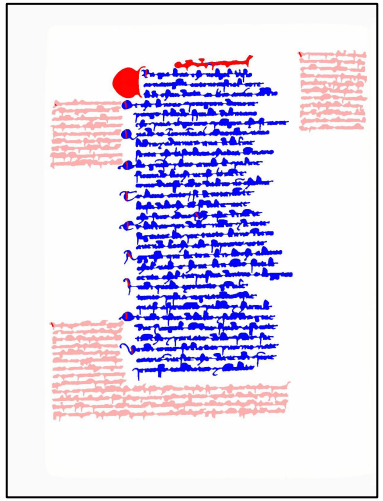


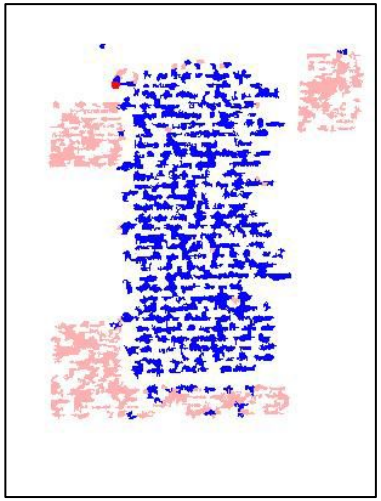


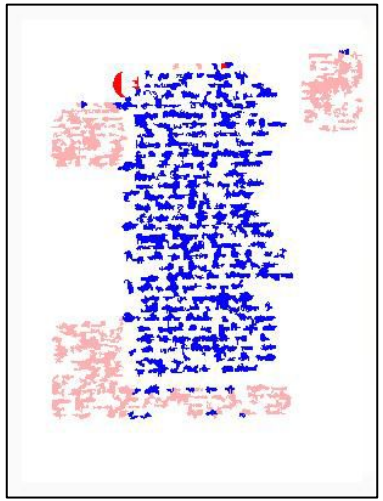
Ground Truth	Local MLP
CRF	CNN

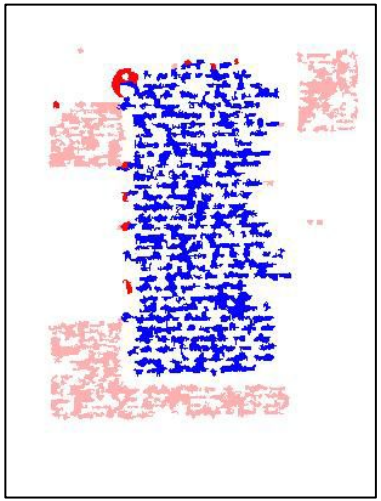


Input

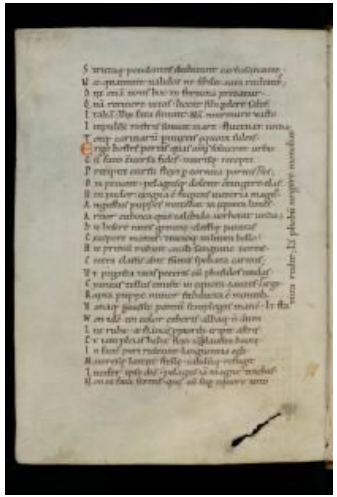




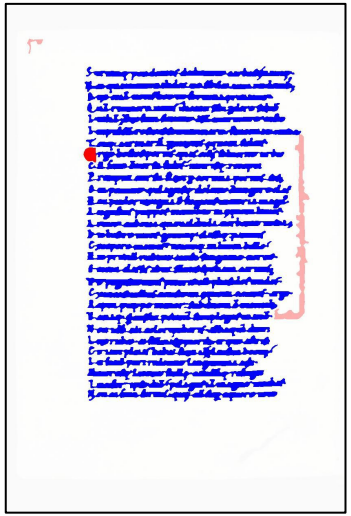


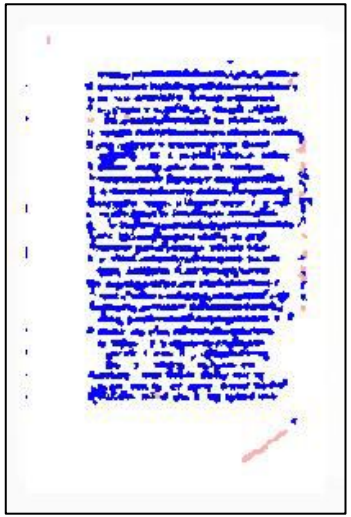


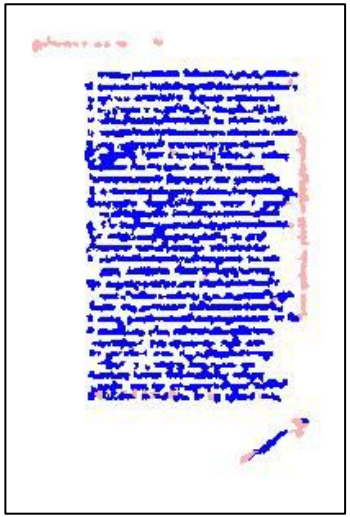
Ground Truth	Local MLP
CRF	CNN



Input

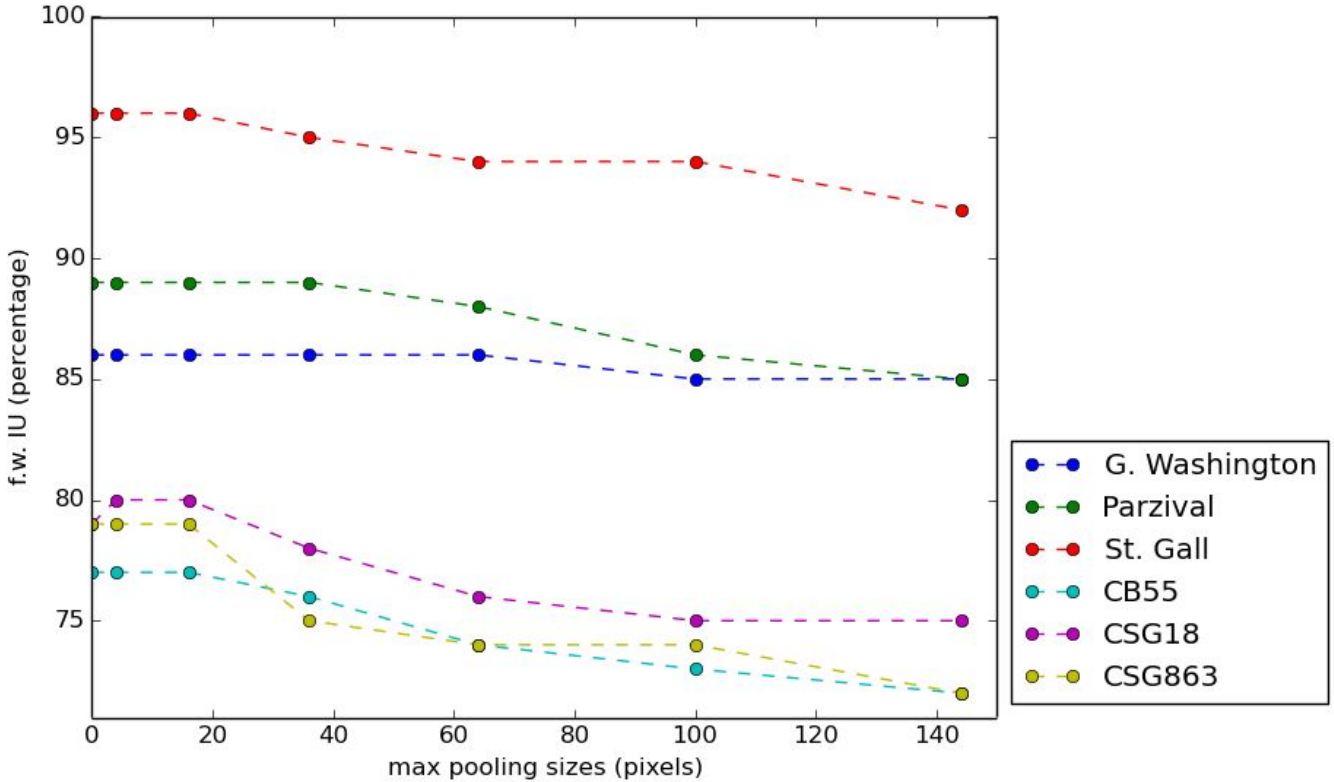




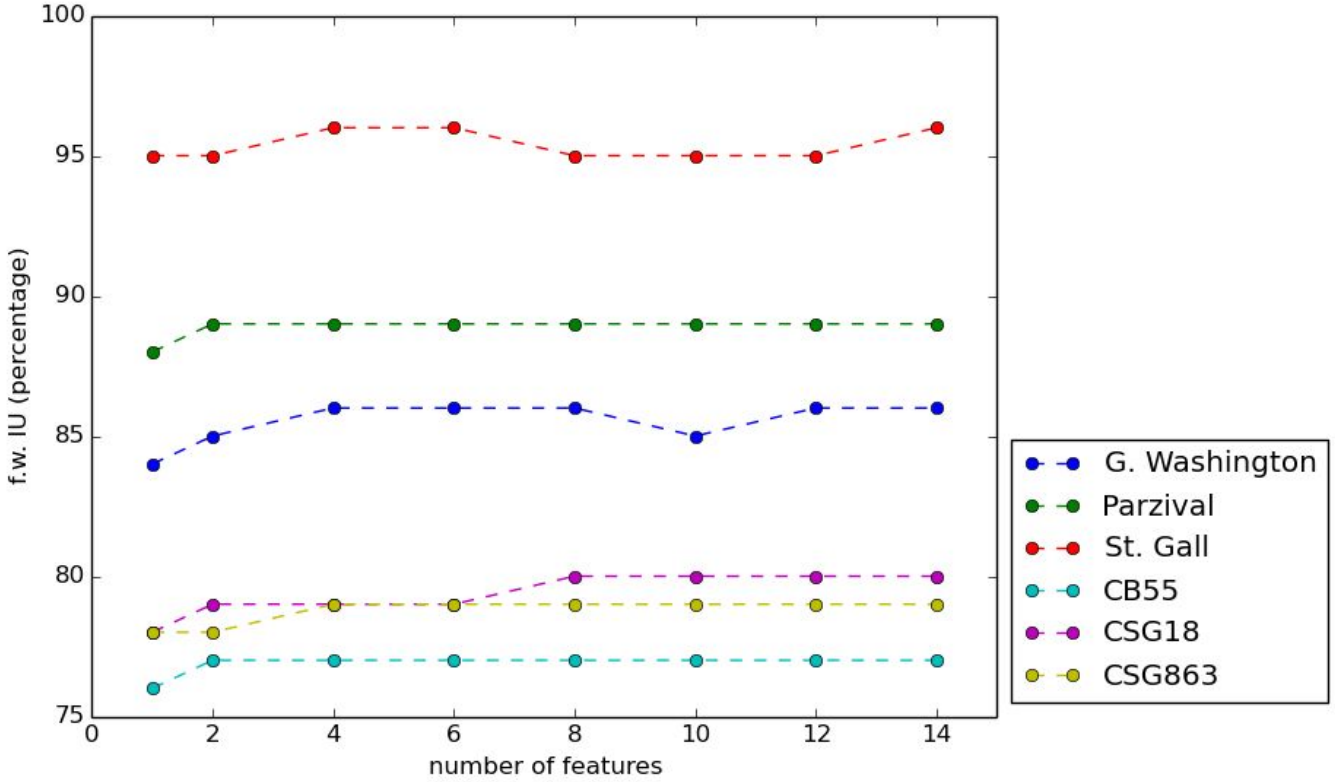


Ground Truth	Local MLP
CRF	CNN

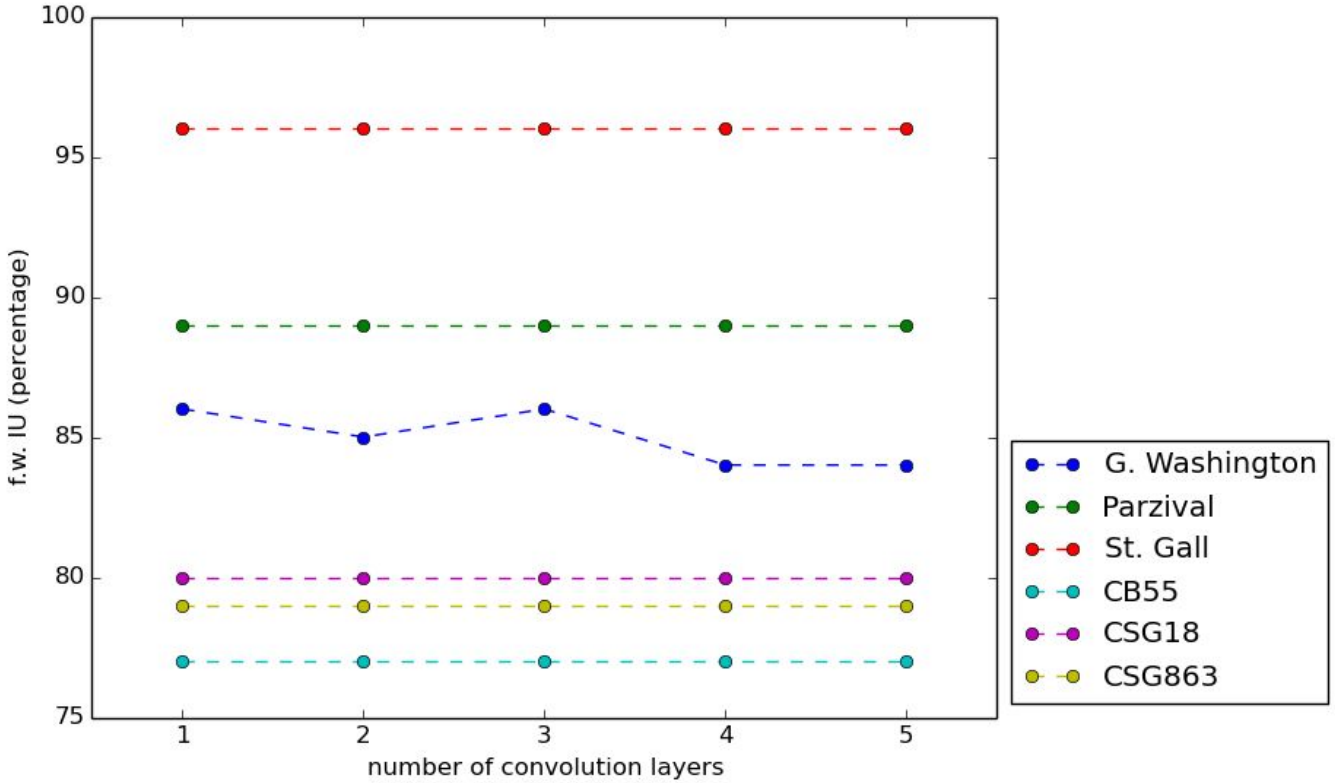
Max Pooling


100
95
90
85
80
75
f.w. IU (percentage)
0
20
40
60
80
100
120
140
max pooling sizes (pixels)
G. Washington
Parzival
St. Gall
CB55
CSG18
CSG863

Number of Features



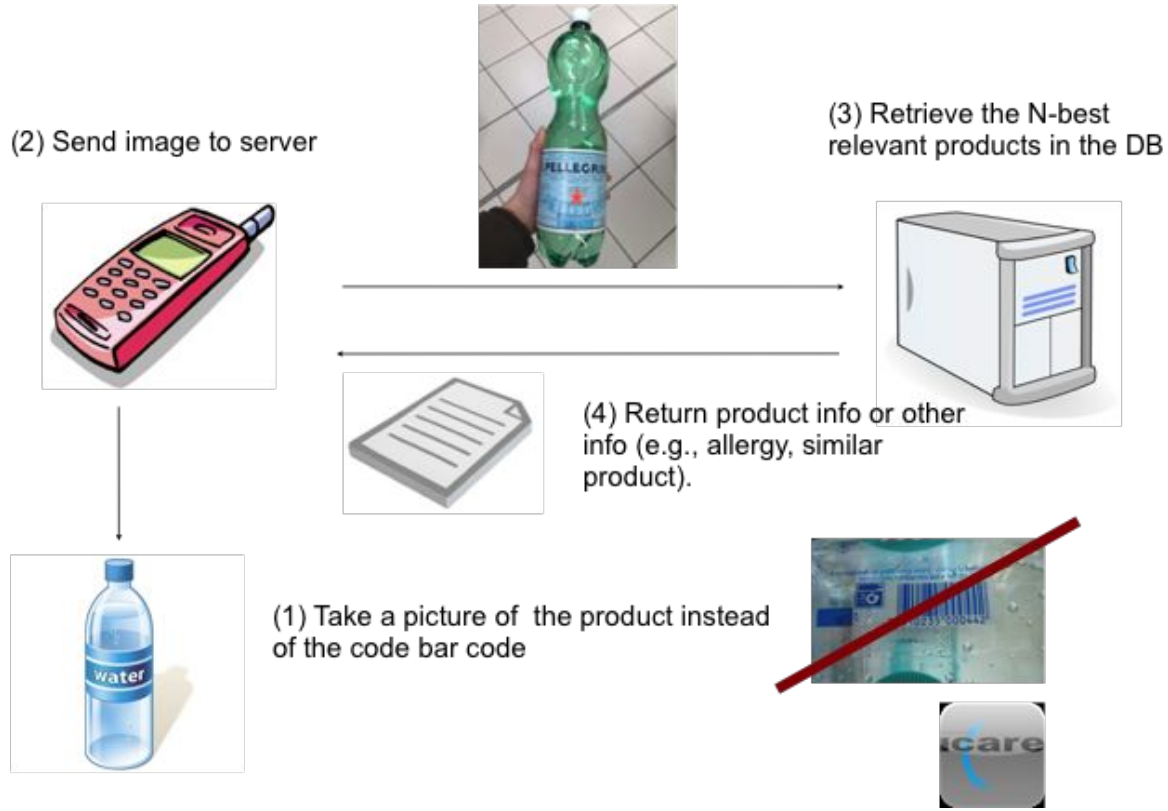
Number of Conv Layers



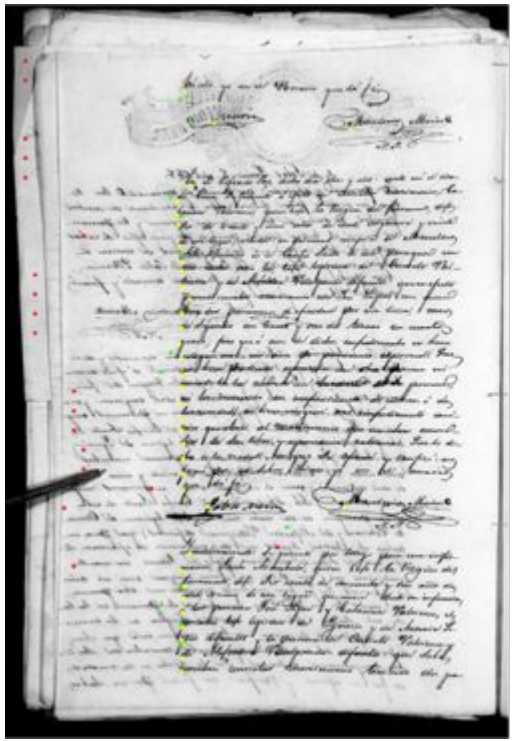
Future Work

- Deep Learning (DL) vs. Probabilistic Graphical Models (PGM)
  - DL: feature learning
  - PGM: modeling the interactions of the elements in a graph
  - How to combine the advantages of DL and PGM ?

Camera-based image retrieval (master project)



Text line detection in historical document

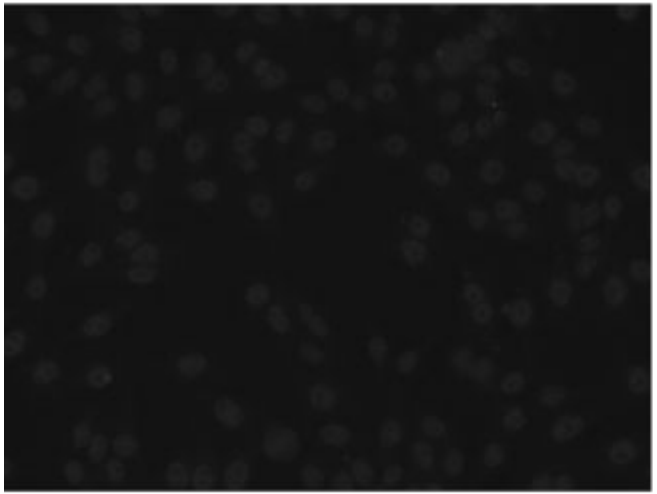


Ground truth

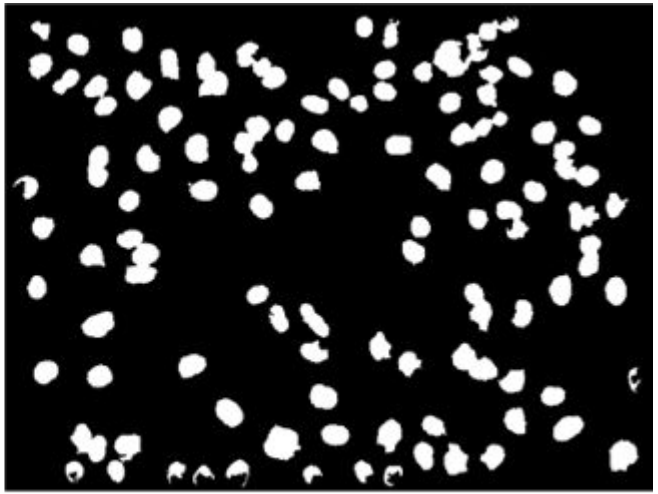
Correct detection

Error

Cell image segmentation



input



output

Other Projects

- Structured Data
  - Booking prediction
  - Revenue per click prediction
  - Consumer shopping prediction
  - Car price prediction
- Computer Vision (CV)
  - Product category classification
- Natural Language Processing (NLP)
  - Toxic Comment Classification (Kaggle challenge, top 16%)
- CV + NLP
  - Product description generation

Github: github.com/ck-unifr