CS 554
Computer Vision

Conditional and Markov Random Fields

Hamdi Dibeklioğlu

Slide Credits: L. van der Maaten

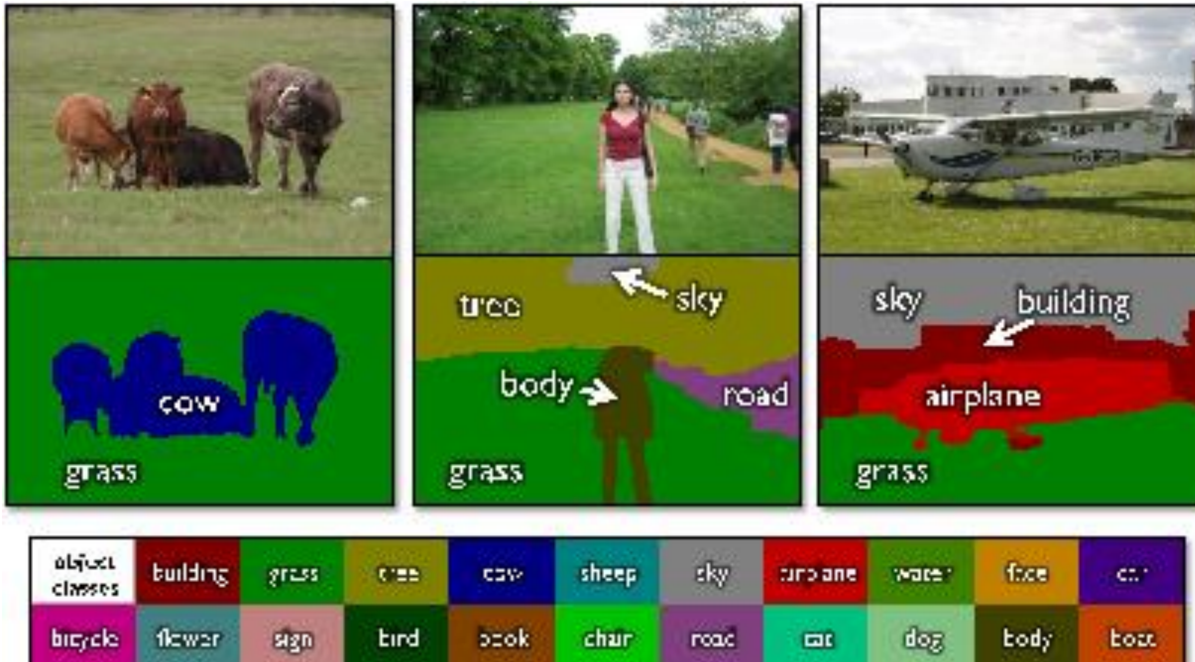
Conditional Random Fields

Supervised image segmentation

- Bounding-box detection is problematic for *articulated objects*

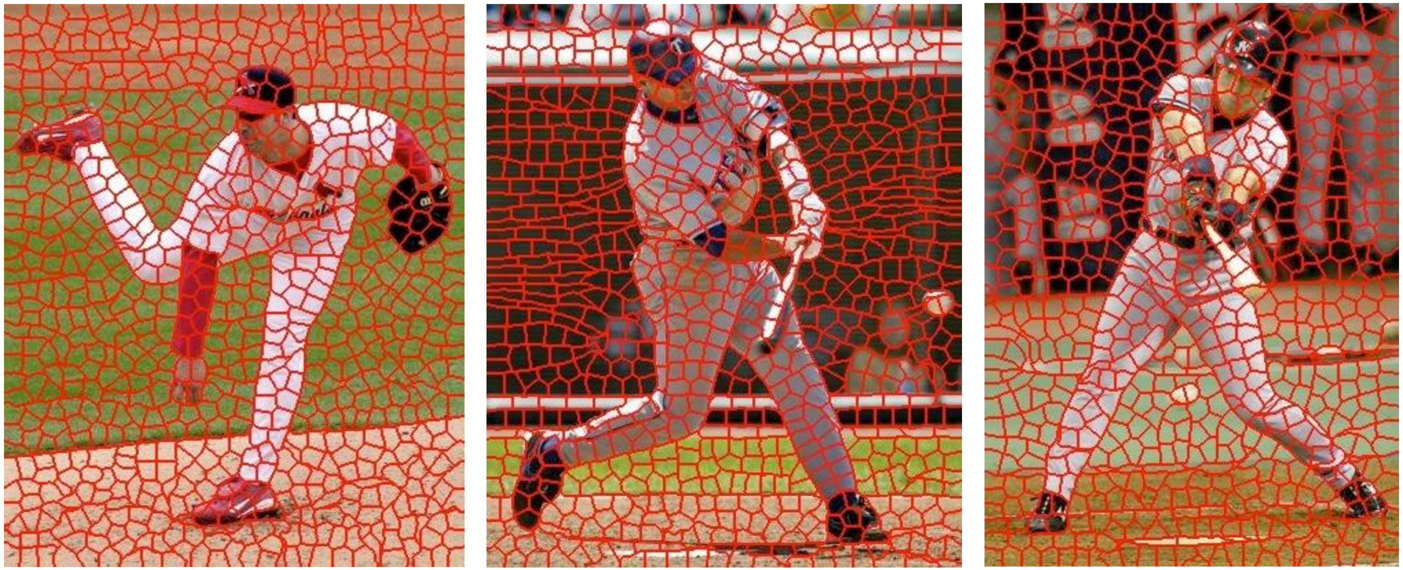
Supervised image segmentation

- Bounding-box detection is problematic for *articulated objects*
- To resolve this issue, we could try to assign a class to each pixel in the image:



Supervised image segmentation

Because images are very large, one often first constructs *superpixels*:



Simple approach to find superpixels: Cluster per-pixel R,G,B,X,Y-features

Supervised image segmentation

Commonly used *features* to represent superpixels:

- Texture layout (textons), color, edge presence, superpixel location, *etc.*

Supervised image segmentation

Commonly used *features* to represent superpixels:

- Texture layout (textons), color, edge presence, superpixel location, *etc.*

Commonly used *classifiers* to assign superpixels to a class:

- Linear classifiers such as *logistic regression* and *support vector machines*
- Ensemble approaches such as *AdaBoost*
- Classifiers that exploit *structure* in the label field (conditional random fields)

Supervised image segmentation

Commonly used *features* to represent superpixels:

- Texture layout (textons), color, edge presence, superpixel location, *etc.*

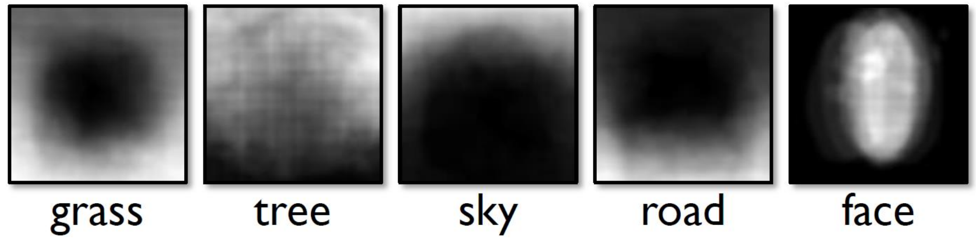
Commonly used *classifiers* to assign superpixels to a class:

- Linear classifiers such as *logistic regression* and *support vector machines*
- Ensemble approaches such as *AdaBoost*
- Classifiers that exploit *structure* in the label field (conditional random fields)

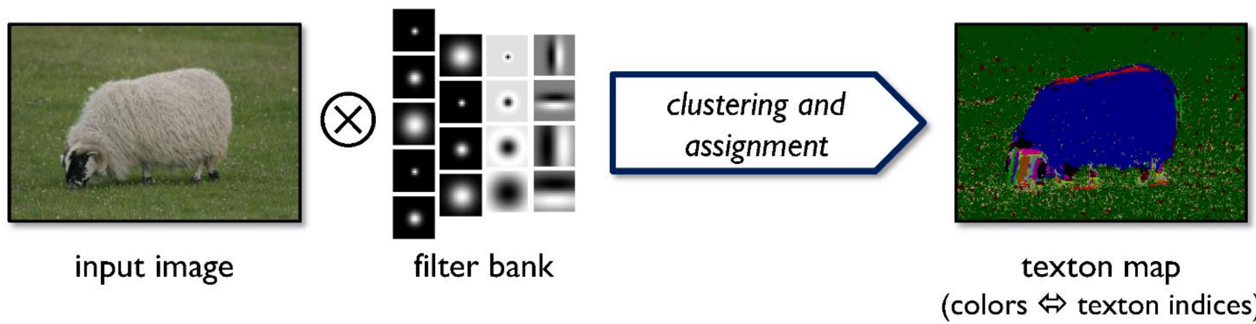
Often, we also incorporate a *location prior* in the segmentation algorithm

Example: TextonBoost

Location:



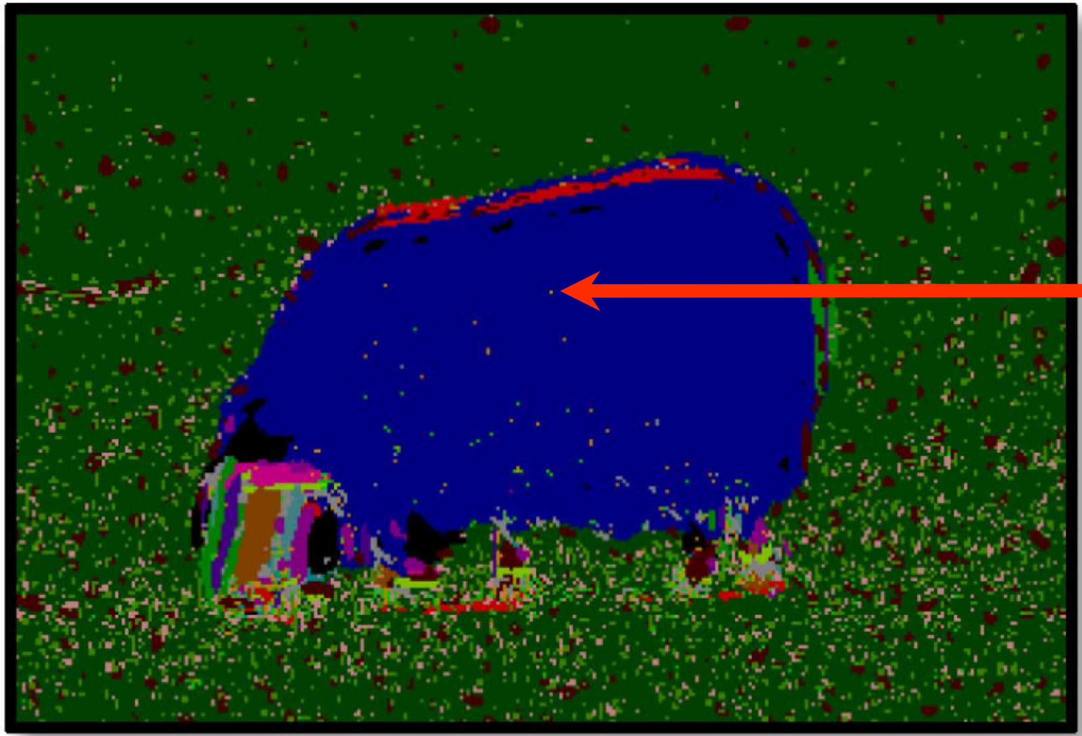
Texture:



Per pixel class posteriors from *texton map* using boosted weak classifiers

Example: TextonBoost

The resulting label image looks quite noisy:



Is this pixel a sheep or not?

Supervised image segmentation

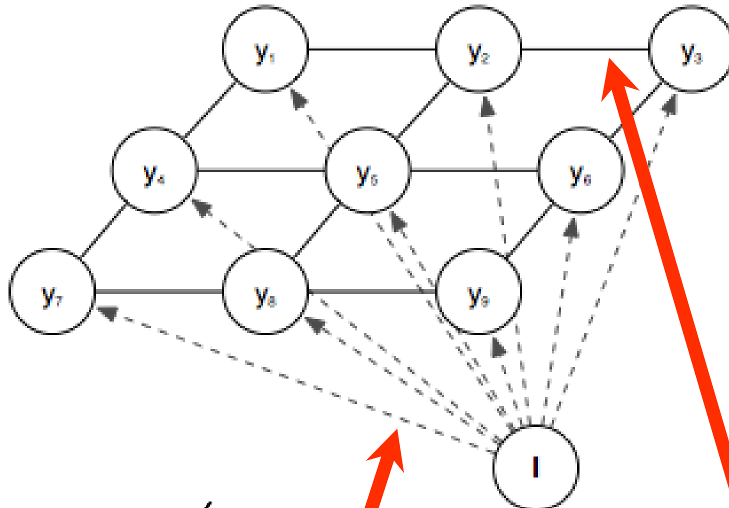
- We know that the *label field* is generally *smooth*: changes are uncommon
- We know that some labels are *incompatible*: “people do not walk on water”

- Conditional and Markov Random Fields allow us to incorporate such things, *e.g.*, to penalize different neighboring labels *except* when there is an edge:

original image

edge potentials

Conditional Random Field



$$p(\mathbf{y}|\mathbf{I}) = \frac{1}{Z(\mathbf{I})} \exp\left(\sum_{i \in V} \Phi(y_i; \mathbf{I}) + \sum_{(i,j) \in E} \Psi(y_i, y_j; \mathbf{I})\right)$$

label field

normalization

exponentiate

score of “regular” classifier

label compatibility function

Edge potential

Example of an edge potential*: $\Psi(y_i, y_j; \mathbf{I}) = \lambda y_i y_j$

Ising model
(encourages similar labeling)

When is an Ising model inappropriate?

- At locations where an image edge is present!

Alternative edge potential: $\Psi(y_i, y_j; \mathbf{I}) = \lambda \exp\left(-\frac{1}{2\sigma^2}(\mathbf{I}_i - \mathbf{I}_j)^2\right) y_i y_j$

- If two pixels are similar, this gives a high penalty for different labels

* Assuming label y is {-1, +1}

Edge potential

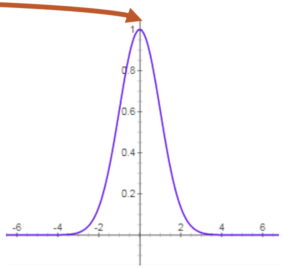
Example of an edge potential*: $\Psi(y_i, y_j; \mathbf{I}) = \lambda y_i y_j$

Ising model
(encourages similar labeling)

When is an Ising model inappropriate?

- At locations where an image edge is present!

Alternative edge potential: $\Psi(y_i, y_j; \mathbf{I}) = \lambda \exp\left(-\frac{1}{2\sigma^2}(\mathbf{I}_i - \mathbf{I}_j)^2\right) y_i y_j$



- If two pixels are similar, this gives a high penalty for different labels

* Assuming label y is {-1, +1}

Inference

Given the CRF model, we need to find the *most likely* labeling (MAP assignment)

We can do this by maximizing the logarithm of the likelihood:

$$\max_{\mathbf{y}} \left[\sum_{i \in V} \Phi(y_i; \mathbf{I}) + \sum_{(i,j) \in E} \Psi(y_i, y_j; \mathbf{I}) - \log Z(\mathbf{I}) \right]$$

Inference

Given the CRF model, we need to find the *most likely* labeling (MAP assignment)

We can do this by maximizing the logarithm of the likelihood:

$$\max_{\mathbf{y}} \left[\sum_{i \in V} \Phi(y_i; \mathbf{I}) + \sum_{(i,j) \in E} \Psi(y_i, y_j; \mathbf{I}) - \cancel{\quad} \right]$$

Inference

Given the CRF model, we need to find the *most likely* labeling (MAP assignment)

We can do this by maximizing the logarithm of the likelihood:

$$\max_{\mathbf{y}} \left[\sum_{i \in V} \Phi(y_i; \mathbf{I}) + \sum_{(i,j) \in E} \Psi(y_i, y_j; \mathbf{I}) - \cdots \right]$$

How many possible labelings are we maximizing over?

Inference

Given the CRF model, we need to find the *most likely* labeling (MAP assignment)

We can do this by maximizing the logarithm of the likelihood:

$$\max_{\mathbf{y}} \left[\sum_{i \in V} \Phi(y_i; \mathbf{I}) + \sum_{(i,j) \in E} \Psi(y_i, y_j; \mathbf{I}) - \cancel{\cdots)} \right]$$

How many possible labelings are we maximizing over?

- For a binary classification problem, there are already two to the power of the number of (super)pixels possible label fields

Inference

Iterated conditional modes (ICM) iteratively maximizes over one of the labels:

$$\max_{y_k} \left[\sum_{i \in V} \Phi(y_i; \mathbf{I}) + \sum_{(i,j) \in E} \Psi(y_i, y_j; \mathbf{I}) - \log Z \right] =$$

Inference

Iterated conditional modes (ICM) iteratively maximizes over one of the labels:

$$\max_{y_k} \left[\sum_{i \in V} \Phi(y_i; \mathbf{I}) + \sum_{(i,j) \in E} \Psi(y_i, y_j; \mathbf{I}) - \log Z \right] =$$

$$\max_{y_k} \left[\Phi(y_k; \mathbf{I}) + \sum_{(k,j) \in E} \Psi(y_k, y_j; \mathbf{I}) \right]$$

Inference

Iterated conditional modes (ICM) iteratively maximizes over one of the labels:

$$\max_{y_k}\left[\sum_{i\in V}\Phi(y_i;\mathbf{I})+\sum_{(i,j)\in E}\Psi(y_i,y_j;\mathbf{I})-\log Z\right]=$$

$$\max_{y_k}\left[\Phi(y_k;\mathbf{I})+\sum_{(k,j)\in E}\Psi(y_k,y_j;\mathbf{I})\right]$$

for a lattice, only need
to compute 5 × #-of-labels terms

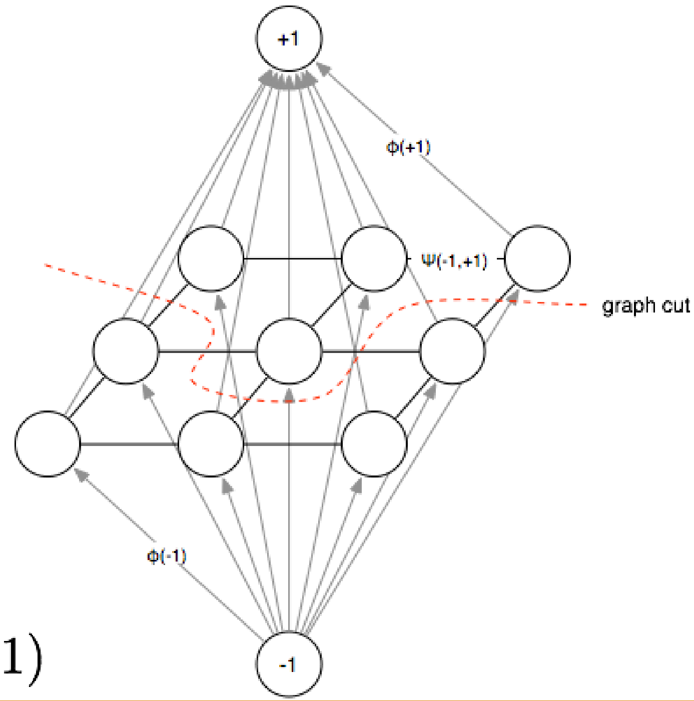
Label field can be initialized to labels that maximize the unary potentials

This procedure converges to a local maximum of the log-likelihood

Inference

MAP solution for *binary pairwise CRF*: $\min_{\mathbf{y}} \sum_{i \in V} \phi(y_i; \mathbf{I}) + \sum_{(i,j) \in E} \psi(y_i, y_j; \mathbf{I})$

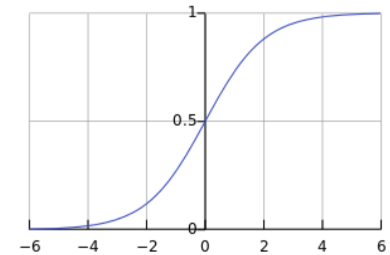
Identical to finding *minimal graph cut* that separates *source* -1 from *sink* 1:



* This assumes that $\psi(-1, +1) = \psi(+1, -1)$

Example: Discriminative Random Fields

Conditional random field model that involves a unary factor:



$$\phi(y_i; \mathbf{I}) = -\log(1 + \exp(-y_i \mathbf{w}^{\mathrm{T}} f_i(\mathbf{I})))$$

model parameters

logistic regressor

image features near site *i*

And a pairwise factor (interaction potential) that is modeled as follows:

$$\psi(y_i, y_j; \mathbf{I}) = K y_i y_j + (1 - K)\left(2 \log\left(\frac{1}{1 + \exp(-y_i y_j \mathbf{v}^{\mathrm{T}} g_{ij}(\mathbf{I}))}\right) - 1\right)$$

logistic regressor
(scaled between -1 and +1)

model
parameters

data-independent term

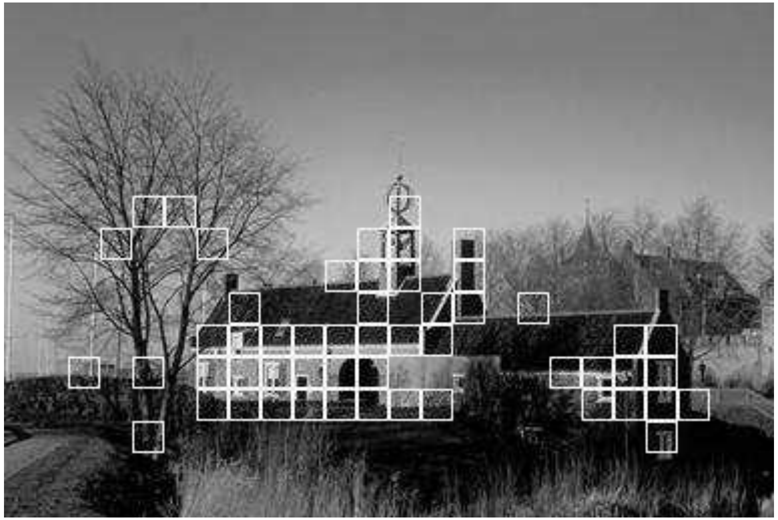
same label?

image features
for pair *(i, j)*

Example: Discriminative Random Fields

The DRF graph is a lattice over neighboring mage patches

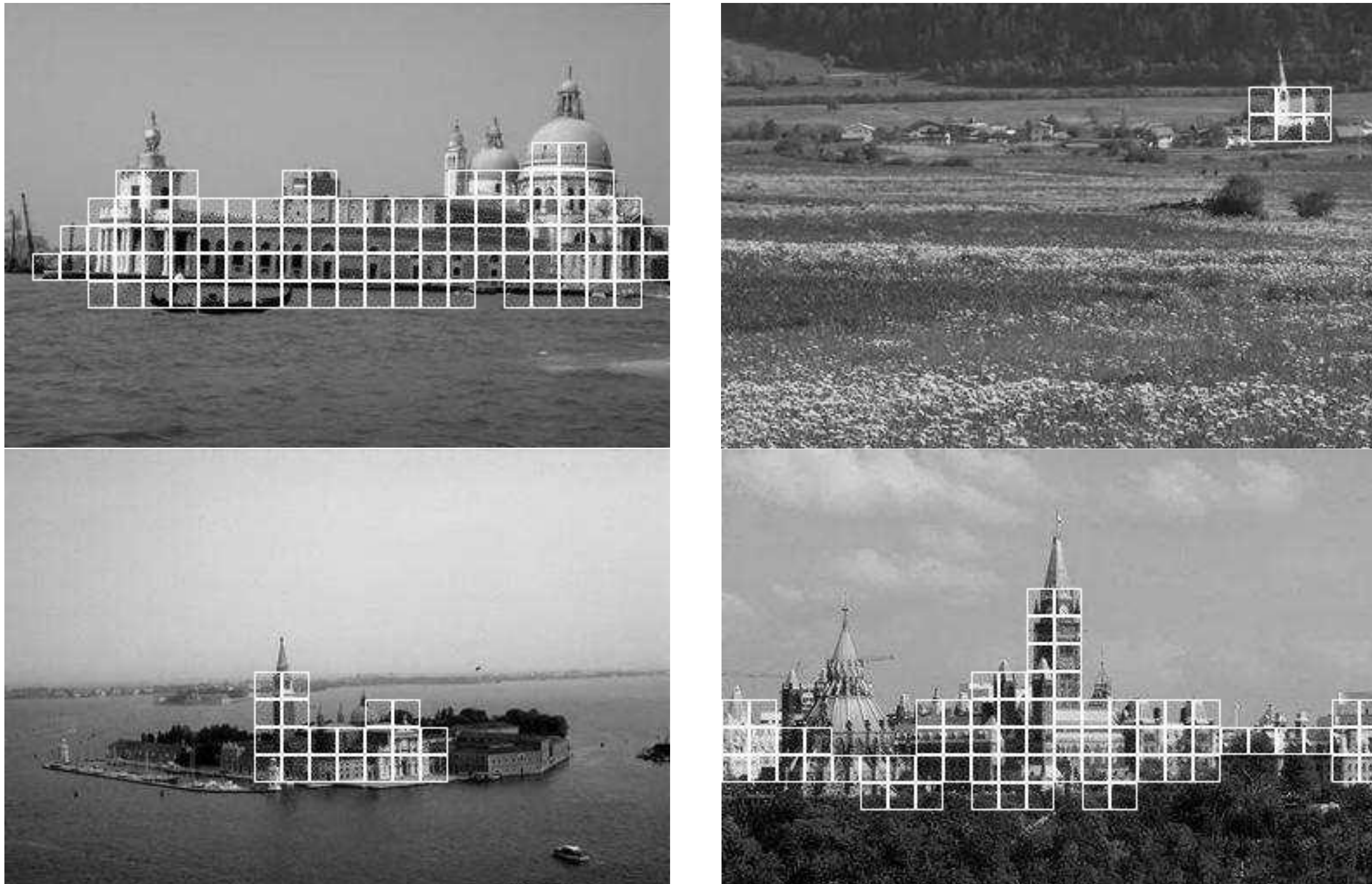
Recognition of “man-made” structures, with and without pairwise factors:



logistic regression

DCRF

Example: Discriminative Random Fields



Example: segmenting occluded people

Markov Random Fields

Markov Random Fields

- In conditional random fields, we defined a distribution of label fields
- In some problems, we want to define a distribution over images $p(\tilde{\mathbf{I}})$:

Markov Random Fields

- In conditional random fields, we defined a distribution of label fields
- In some problems, we want to define a distribution over images $p(\tilde{\mathbf{I}})$:
  - Assume our image is corrupted by Gaussian noise
  - We can then try to infer the non-corrupted image by maximizing:

$$p(\tilde{\mathbf{I}}|\mathbf{I}) \propto p(\mathbf{I}|\tilde{\mathbf{I}})p(\tilde{\mathbf{I}})$$

non-corrupted image

observed, corrupted image

corruption model (Gaussian noise)

prior over images

  - Markov Random Fields are appropriate model for $p(\tilde{\mathbf{I}})$

Markov Random Fields

- An example of a Markov Random Field is the following model:

$$p(\tilde{\mathbf{I}}) = \frac{1}{Z} \exp\left(\sum_{i \in V} \Phi(\tilde{\mathbf{I}}_i) + \sum_{(i,j) \in E} \Psi(\tilde{\mathbf{I}}_i, \tilde{\mathbf{I}}_j) \right)$$

- Key difference with CRFs: we do not *condition* on the image
- This makes inference in MRFs is even harder than in CRFs. Why?
  - MRFs need to normalize over *all possible images* instead of all possible labelings
  - However, similar inference algorithms as before are generally be applied

Example: Simple denoising MRF

- Example of using a simple MRF over binary (-1, +1) images for denoising:

$$p(\tilde{\mathbf{I}}|\mathbf{I}) \propto p(\mathbf{I}|\tilde{\mathbf{I}})p(\tilde{\mathbf{I}})$$

$$p(\tilde{\mathbf{I}}|\mathbf{I}) \propto \overbrace{\exp\left(\eta \sum_{i \in V} \tilde{\mathbf{I}}_i \mathbf{I}_i\right)}^{\text{Corruption model}} \overbrace{\exp\left(\alpha \sum_{i \in V} \tilde{\mathbf{I}}_i + \beta \sum_{(i,j) \in E} \tilde{\mathbf{I}}_i \tilde{\mathbf{I}}_j\right)}^{\text{MRF prior}}$$

“cost” for changing a pixel value

prior over individual pixel

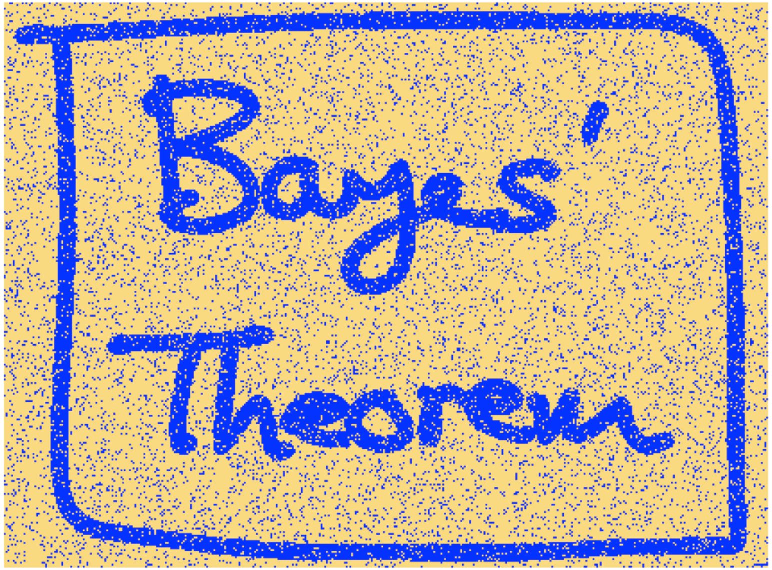
prior over neighboring pixels

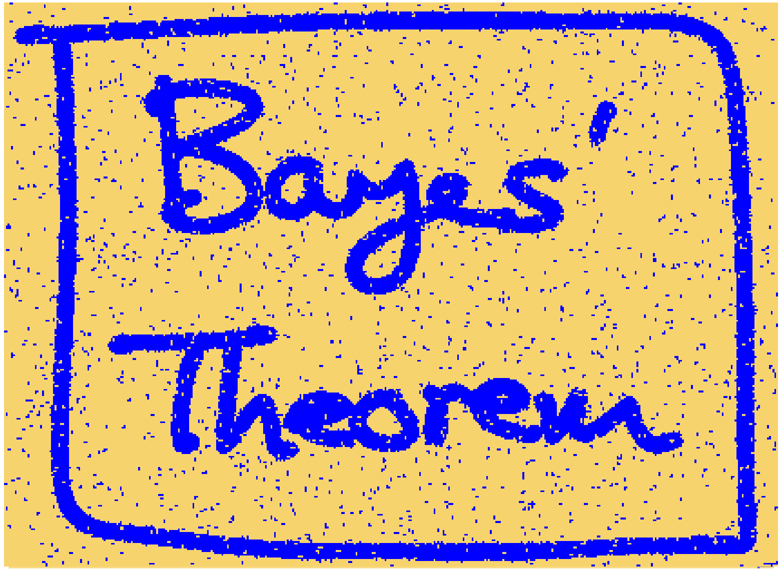
- Note: MAP inference for this simple MRF is similar to the simple CRF earlier

Example: Simple denoising MRF

- Example of using a simple MRF over binary (-1, +1) images for denoising:

$$p(\tilde{\mathbf{I}}|\mathbf{I}) \propto \exp\left(\eta \sum_{i \in V} \tilde{\mathbf{I}}_i \mathbf{I}_i\right) \exp\left(\alpha \sum_{i \in V} \tilde{\mathbf{I}}_i + \beta \sum_{(i,j) \in E} \tilde{\mathbf{I}}_i \tilde{\mathbf{I}}_j\right)$$



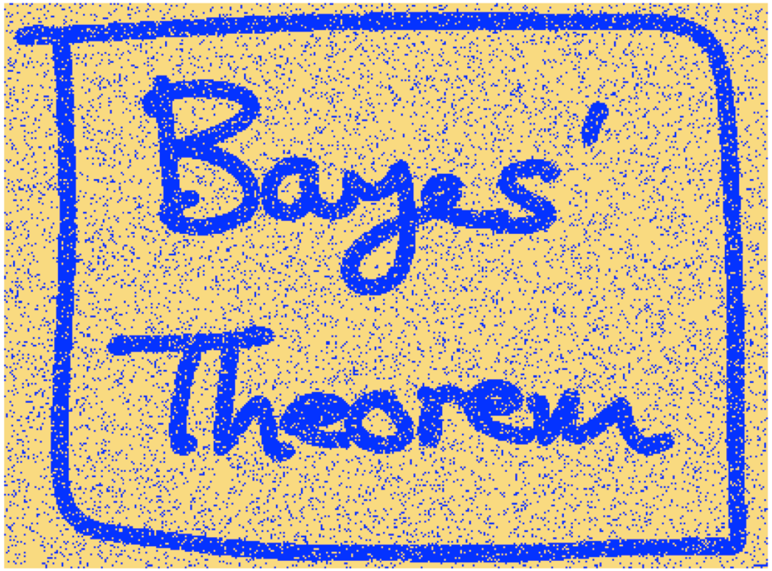


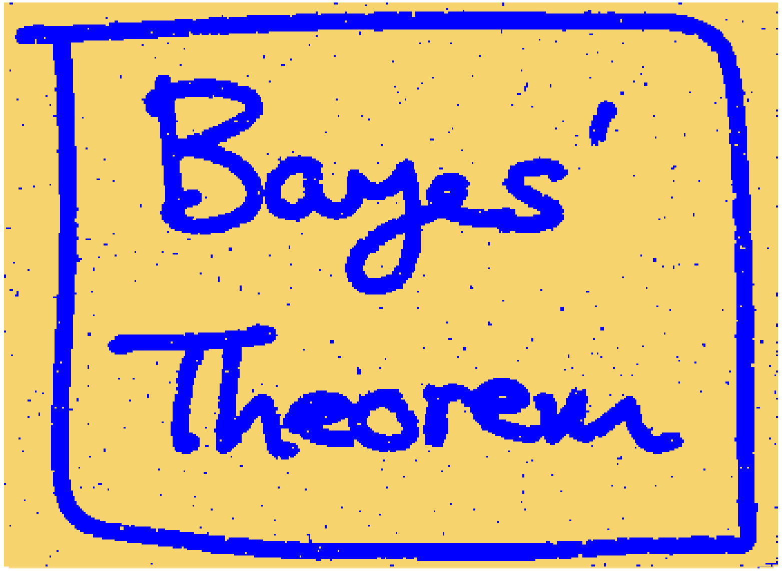
Graph Cut (MAP)

Example: Simple denoising MRF

- Example of using a simple MRF over binary (-1, +1) images for denoising:

$$p(\tilde{\mathbf{I}}|\mathbf{I}) \propto \exp\left(\eta \sum_{i \in V} \tilde{\mathbf{I}}_i \mathbf{I}_i\right) \exp\left(\alpha \sum_{i \in V} \tilde{\mathbf{I}}_i + \beta \sum_{(i,j) \in E} \tilde{\mathbf{I}}_i \tilde{\mathbf{I}}_j\right)$$



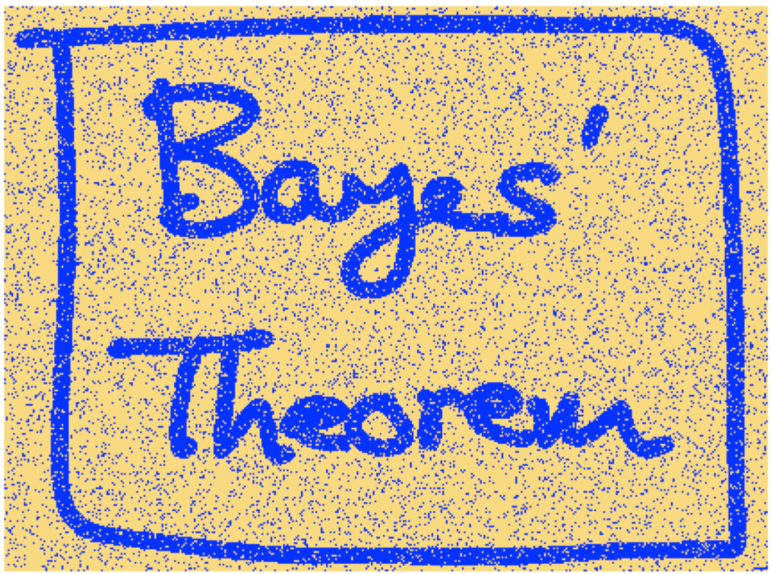


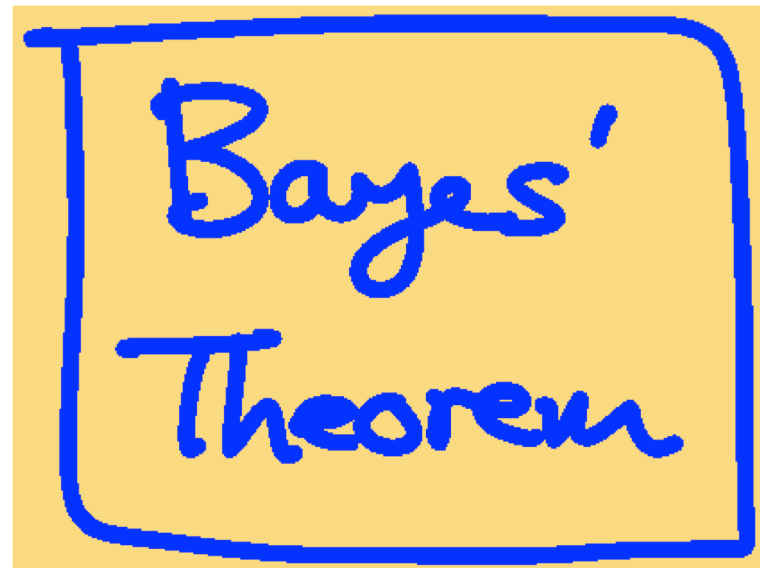
Graph Cut (MAP)

Example: Simple denoising MRF

- Example of using a simple MRF over binary (-1, +1) images for denoising:

$$p(\tilde{\mathbf{I}}|\mathbf{I}) \propto \exp\left(\eta \sum_{i \in V} \tilde{\mathbf{I}}_i \mathbf{I}_i\right) \exp\left(\alpha \sum_{i \in V} \tilde{\mathbf{I}}_i + \beta \sum_{(i,j) \in E} \tilde{\mathbf{I}}_i \tilde{\mathbf{I}}_j\right)$$





Graph Cut (MAP)

Example: Fields of Experts

- FoE models each potential using a *product of Student-t distributions*:

$$p(\mathbf{I};\Theta) = \frac{1}{Z(\Theta)} \exp\left(\sum_{k=1}^{K} \sum_{n=1}^{N} \log \left(1 + \frac{1}{2}(\mathbf{J}_n^{\mathrm{T}} \mathbf{I}_{(k)})^2 \right)^{-\alpha_i} \right)$$

sum over
multiple filters
and patches

log of Student-t distribution
(heavy-tailed distribution)

filtered *k*-th
image patch $I_{(k)}$

- Intuitively, the model assigns a probability to an image as follows:
  - Patch gets high probability if it does not look like any of the filters (zero inner product)
  - Image gets high probability if many of the patches get high probability

Example: Fields of Experts

- Learning expert filters independently vs. within Markov Random Field
- Train experts on generic image database
- **Q: Why will we not learn trivial filters that are all zero?**

Using independent image patches

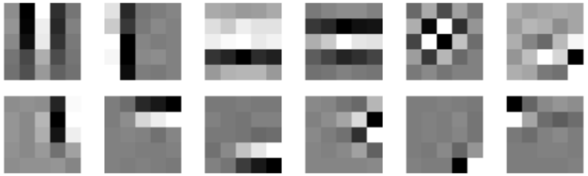
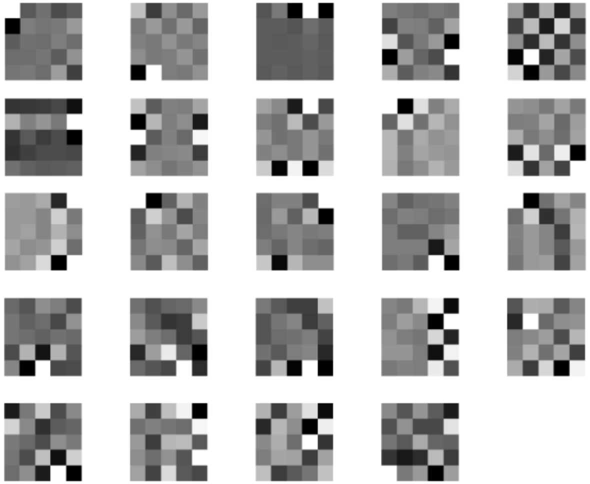


Image patches as potentials in MRF



MRF has
no clear structure?

Filters account for statistical dependency
in overlapping patches?

Denoising using FoE

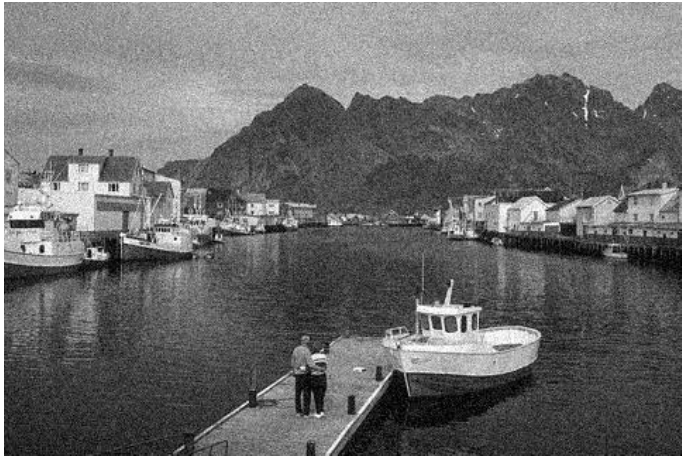
- Using the FoE as image prior, denoising can be phrased as a MAP-problem:

$$\max_{\tilde{\mathbf{I}}} \mathcal{N}(\mathbf{I}|\tilde{\mathbf{I}}, \sigma^2)p(\tilde{\mathbf{I}})$$

corruption model
(Gaussian noise)

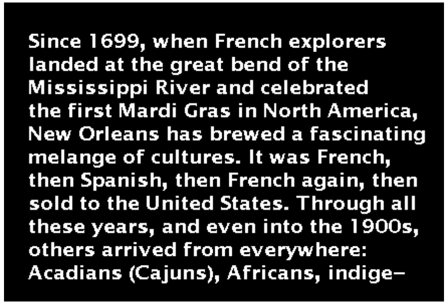
field of experts prior

- The result of the MAP-inference has removed Gaussian noise from the image:



Inpainting using FoE

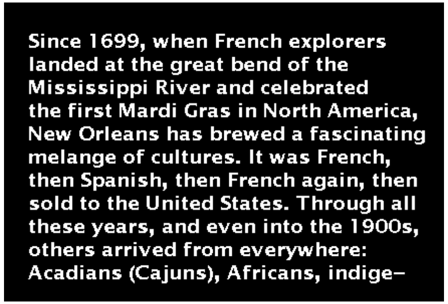
- Given a mask image, inpainting can also be phrased as a MAP-problem:



- Example of inpainting to remove text from an image:

Inpainting using FoE

- Given a mask image, inpainting can also be phrased as a MAP-problem:



- Example of inpainting to remove text from an image:

Inpainting using FoE

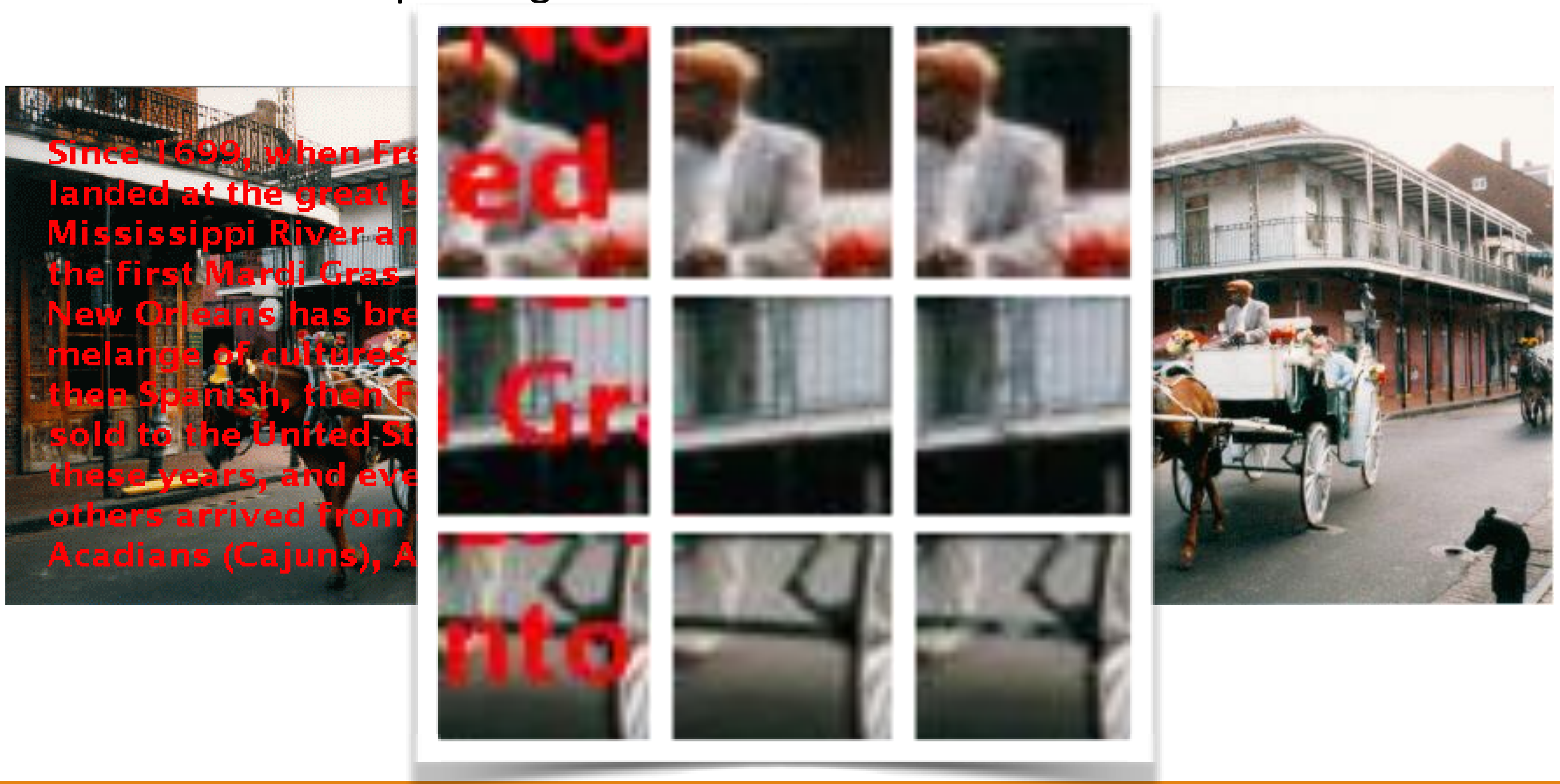
Closer look of the inpainting results:

Inpainting using FoE

Closer look of the inpainting results:

Inpainting using FoE

Closer look of the inpainting results:



Inpainting using FoE

Closer look of the inpainting results:

Can you give an intuition for what the FoE model has learned?

- Hard to say, but for instance: Edges generally continue in same direction

Reading material:

- Section 3.7 and 10.5 and Appendix B of Szeliski
- S. Kumar, and M. Hebert. "Man-made structure detection in natural images using a causal multiscale random field." CVPR, 2003.
- Roth, Stefan, and Michael J. Black. "Fields of experts." *International Journal of Computer Vision* 82, no. 2 (2009): 205.