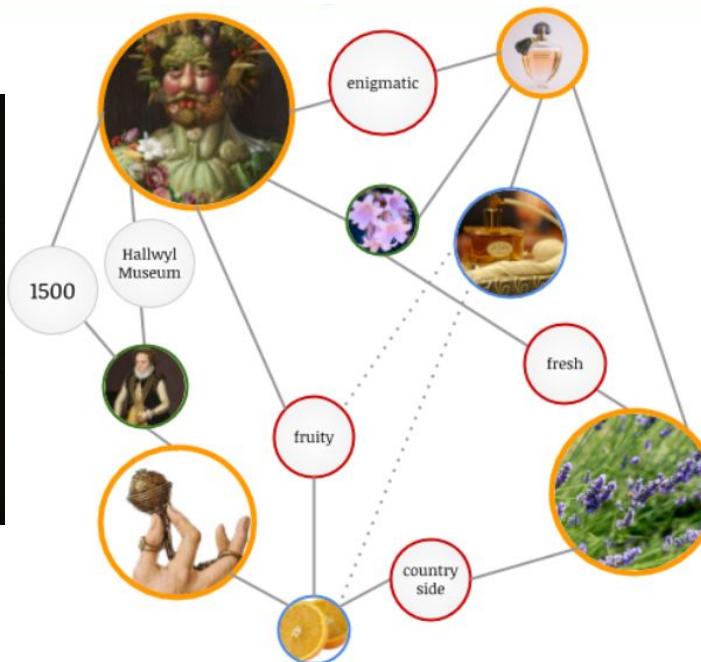
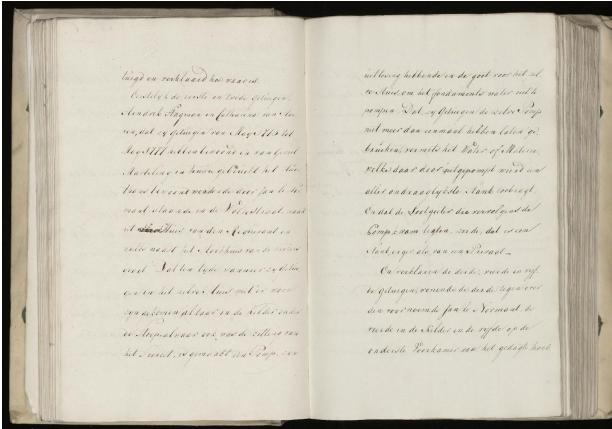


Visual Recognition for the Humanities

Mathias Zinnen,
Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg
WiSe 2023/24



ML and the Humanities



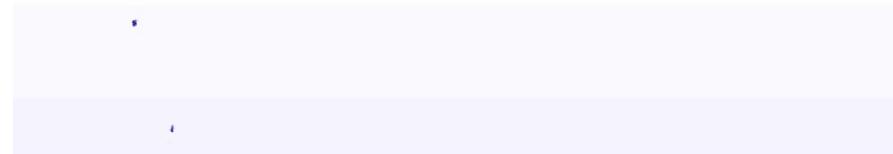
Focus I - Text

Text and Writer identification
and OCR

Handwriting imitation

Α' τὸ τῆς Φυγῆς τῇ Σίρξε ἐκ τῆς Ελλάδος ἔστως τῆς ἡ Μυκάλη γίνεται.

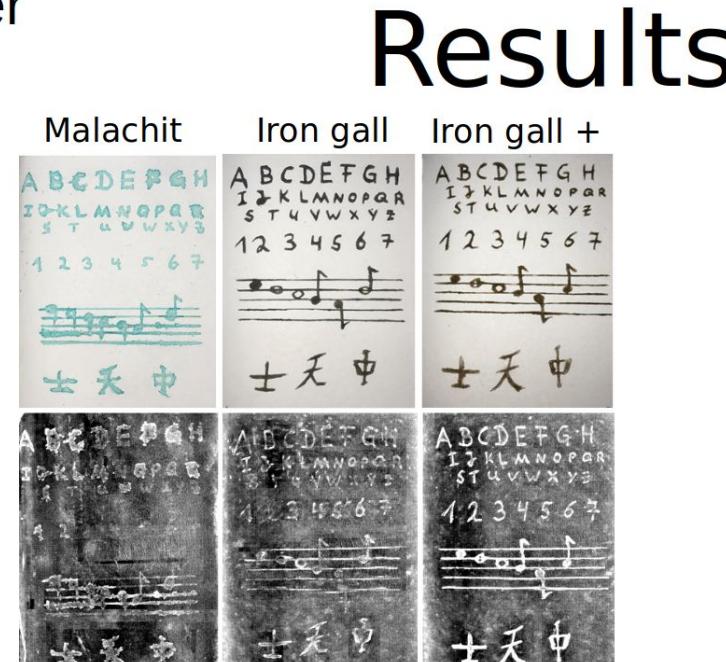
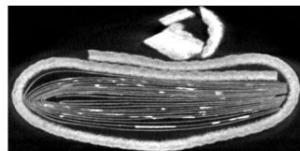
Hη πρώτη φάμετος τῶν Ελλήνων Φροντίς μετὰ την οὐρανοπομάχην, ὃς τὸν οὐρανόν εἰσελιώσιν εἰς τὰ Δελφούς τὰς ἀπαρχὰς τῶν πλεονὸν λαθύρων, ὅσα ἔλαβον ἀπὸ τῆς Πέρσεως. Ως σύμμαχοι θεοφρόμενοι οἱ Ἕλληνες ἤστιν πάντοτε προτεκτικοὶ εἰς τὰ καθήκοντα τῆς Θρησκείας, οἱ μὲν ὅλοι ὅτι αἱ αἱρέσεις οὐ τὰ δόγματα τῶν φιλοσόφων ἐλίδασκον τὸν αἰθρόπτινον γένος ἀλίγους οὐ τιμᾶ τὰς δημοσίες τελετὰς, οὐ τόσους ἡ Θρησκεικοὺς ὅτινος πρόστοντος οἱ μίσιοι σύνδεσμοι τῆς ἑπούσεως αὐτῶν, οὐ ἐκράτει αὐτὸς τῷος ὥρας ἀδιναστώς ἡμαρτυρεῖν. ὅτε δὲ ἐλύθη αὐτὸς ὁ δεσμός, οὐ οἱ Αἱμφίκτιοι καὶ σύνδος ἔγινε τελτικὴ μᾶλλον ἡ θεοκρετικὴ σύνδος, διελύθη οὐ γενικὴ ἔνωσις, οὐ πολλαὶ ἐπαρχίαι ἔγιναν θύμα τῶν ἐμφυλίων πολέμων.



Focus II - Book CT [1]



Daniel Stromer



[1] Stromer, Daniel; Christlein, Vincent; Huang, Xiaolin; Zippert, Patrick; Hausotte, Tino; Maier, Andreas [Virtual cleaning and unwrapping of non-invasively digitized soiled bamboo scrolls](#) Scientific Reports, vol. 9, no. 1, pp. 23111
2019

Focus III - Similarity [2]



GERMANISCHES
NATIONAL
MUSEUM

Aline Sindel

Image Registration

Portraits



Prints



Similarity Analysis

Repeating Elements



Multi-modal



Mixed Type



Repeating Motives



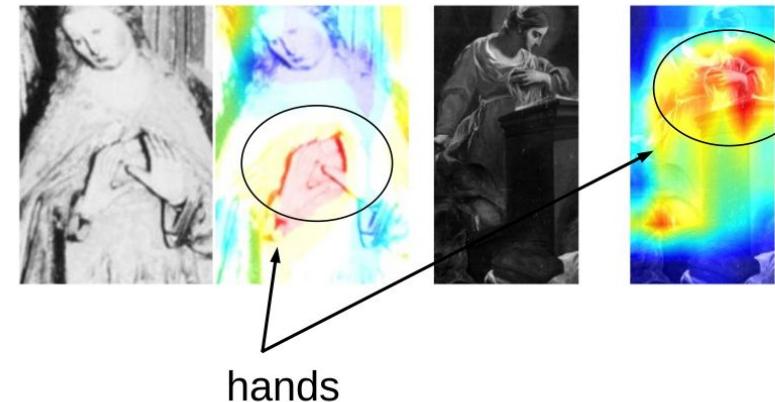
[2] Project page : <https://www.gnm.de/forschung/projekte/luther-bildnisse/>

Focus IV - Understanding [3]



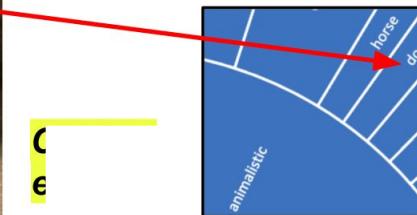
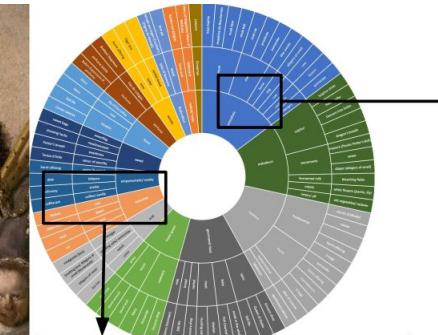
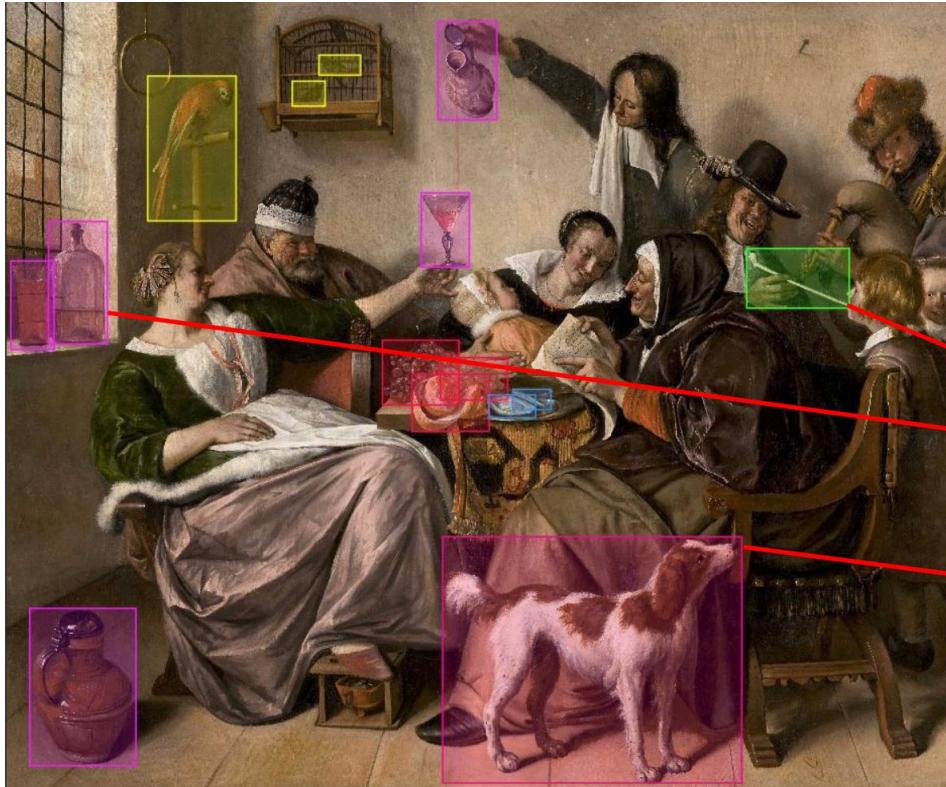
Ronak Kosti

Region of Interest
for Mary



[3] Madhu, Prathmesh, et al. "Recognizing Characters in Art History Using Deep Learning." *Proceedings of the 1st Workshop on Structuring and Understanding of Multimedia heritAge Contents.* 2019.

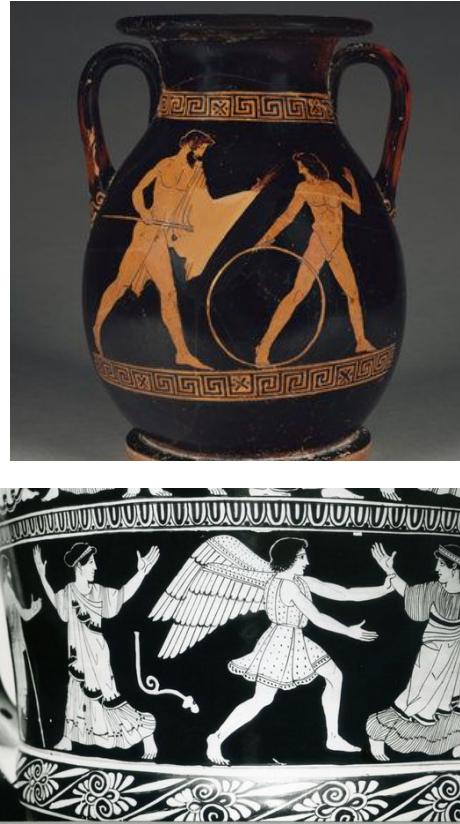
Focus V - Digital Heritage



Christian Iconography



Classical Arch.



Christian Arch.



Visual recognition?



Visual recognition?

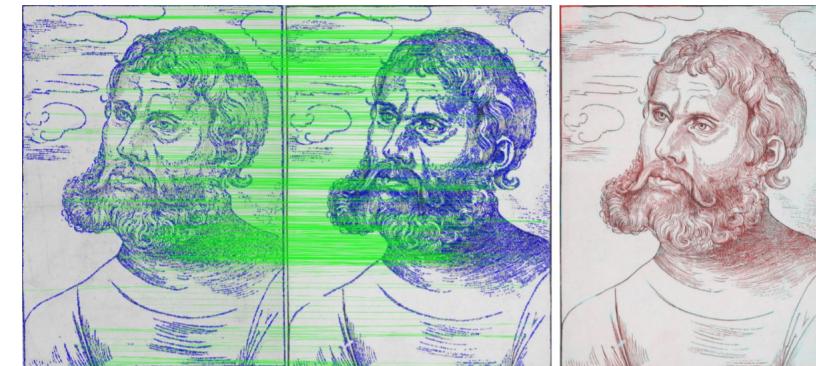


Visual recognition?

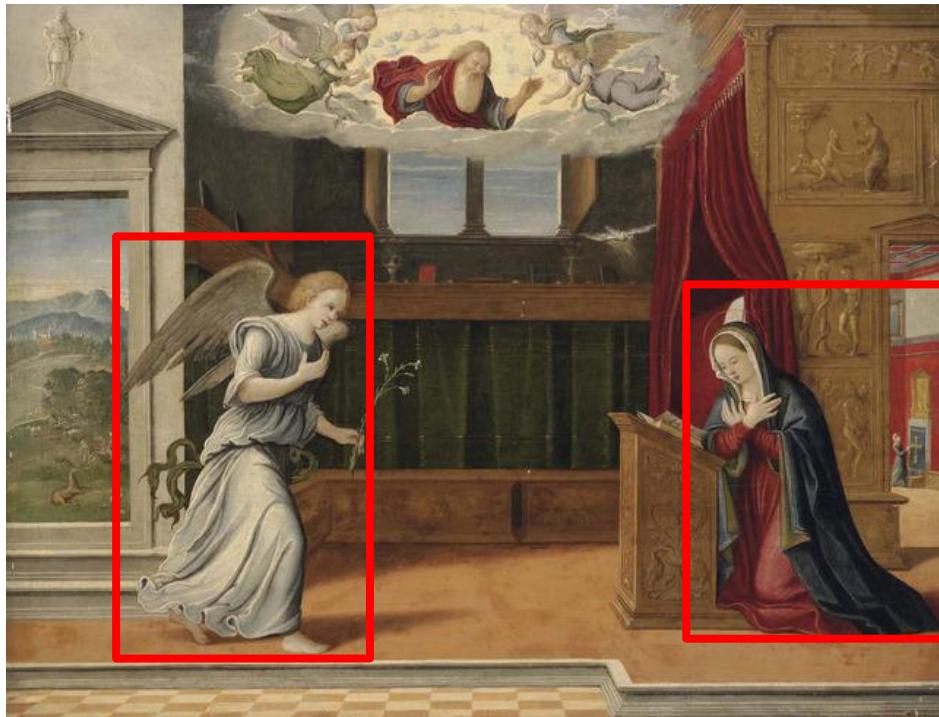


Visual recognition?

- Classification
- Object Detection
- Semantic Segmentation
- Instance Segmentation
- Key Point Detection
- VQA

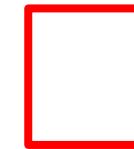


Visual recognition?

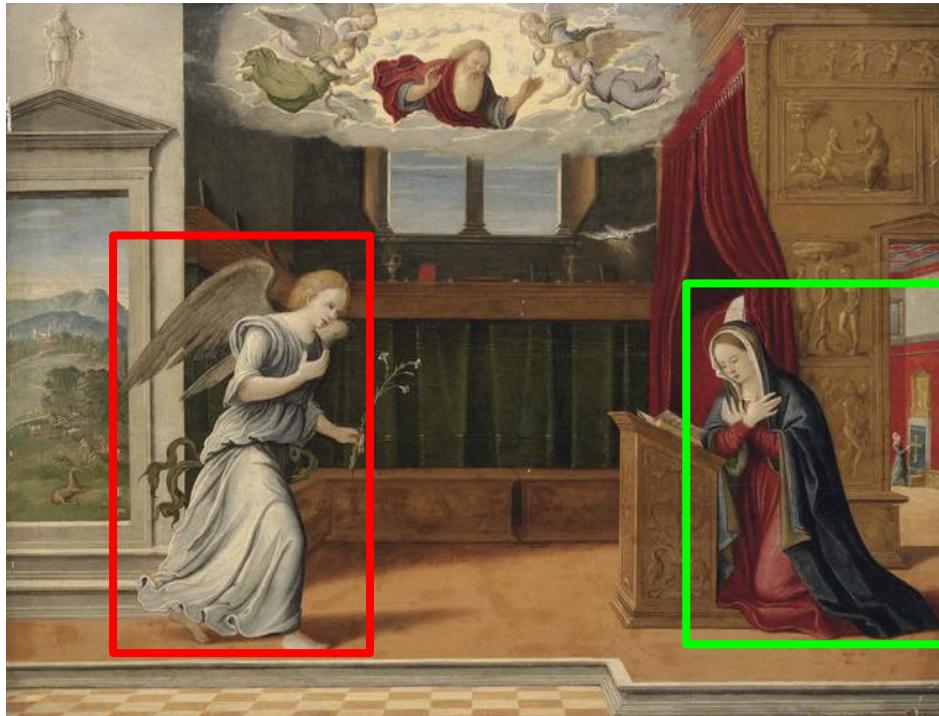


Category-level Recognition

Persons



Visual recognition?



Instance-level Recognition

Mary

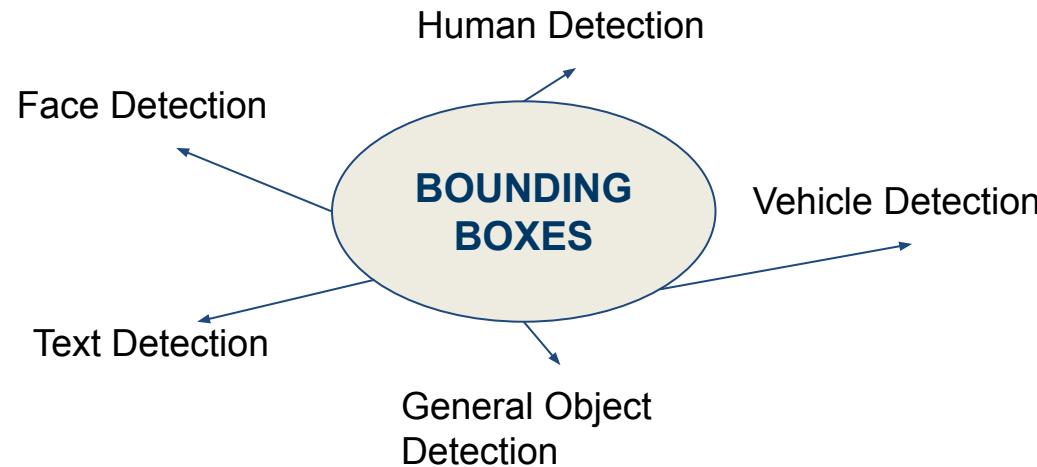


Gabriel



How to quantify objects?

- Encapsulating objects → a common representation



Quantifying objects - Applications

- Cancer detection in radiology-based images [Healthcare]
 - Autonomous Driving [Automation]
 - Detection of manufacturing defects, factory floor surveillance [Manufacturing]
 - Pedestrian Detection [Public safety and surveillance]
- ... and many more ...

Modern Object Detection

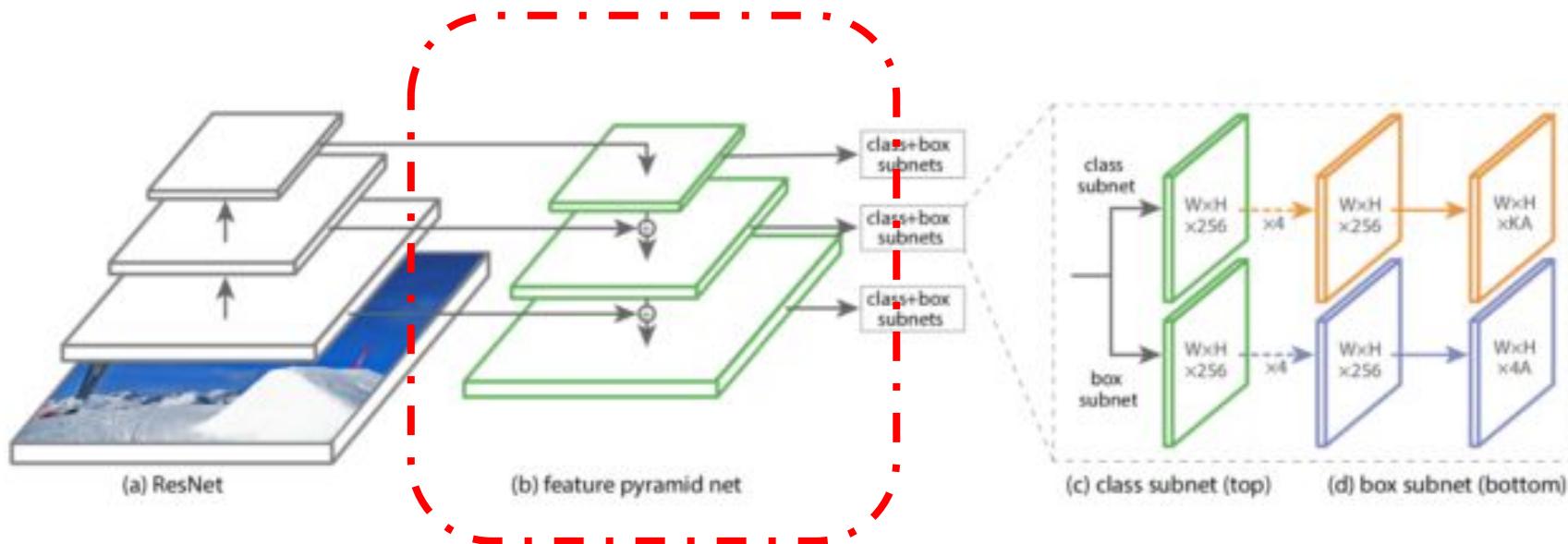
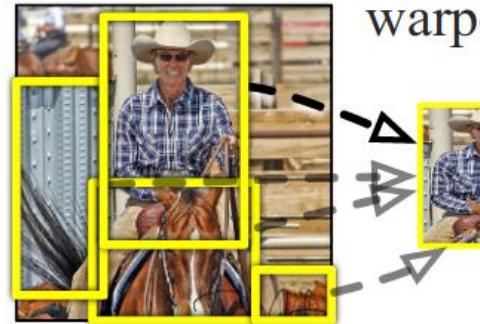


Figure - RetinaNet Architecture

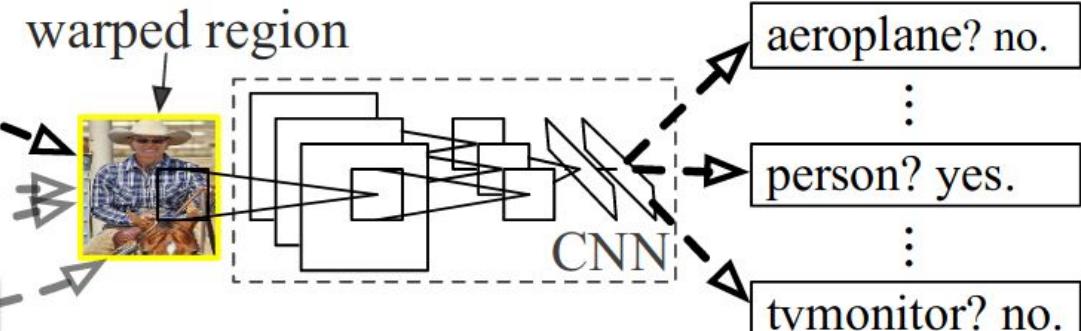
Modern Object Detection



1. Input
image



2. Extract region
proposals (~2k)



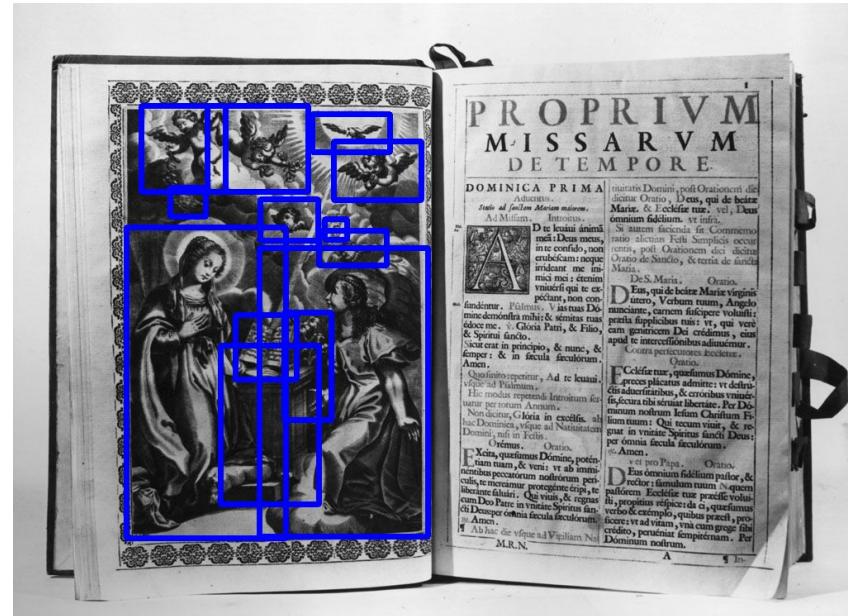
3. Compute
CNN features

4. Classify
regions

Figure - R-CNN Architecture

Challenges

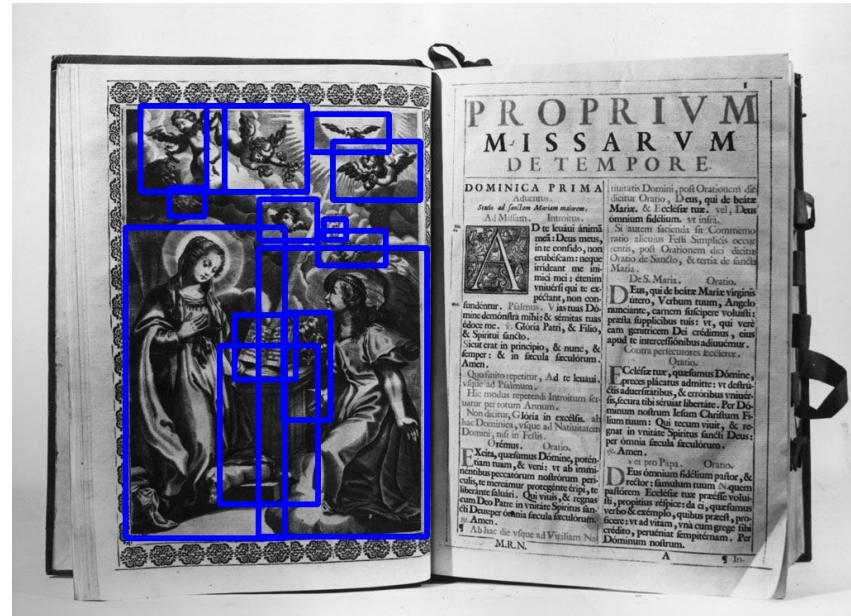
- Existing methods. Domain shift?
- Amount of Annotations
- What about new classes?



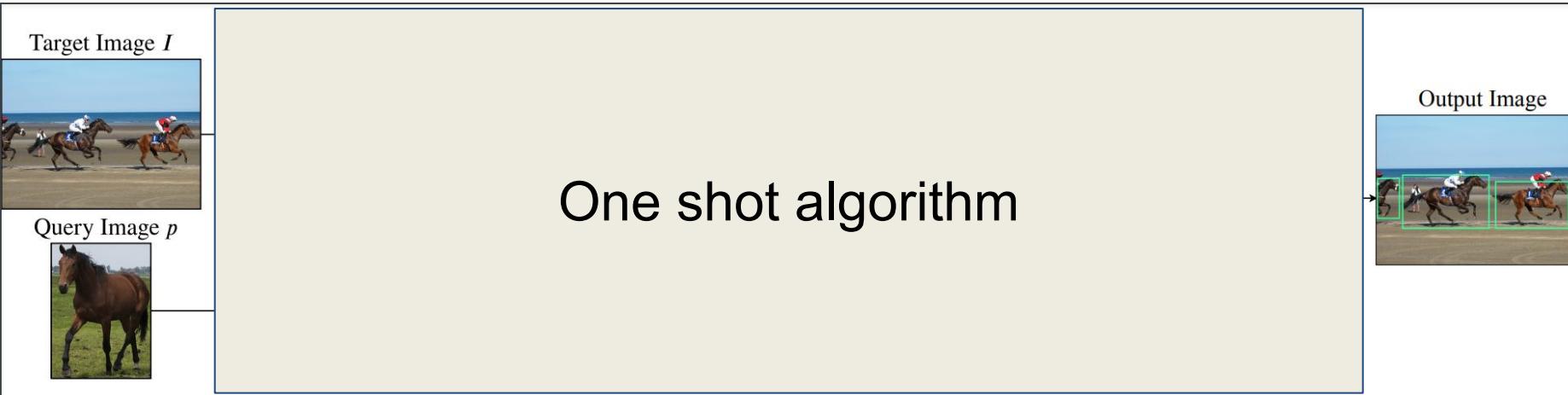
Challenges

- Existing methods. Domain shift?
- Amount of Annotations
- What about new classes?

N-Way K-Shot Object Detection



One-Shot Object Detection



- Train/Seen classes – [Horses, hats, cats, donkeys]
- Test/Unseen classes – [Persons, dogs]

What do these deep features encode? OR

How do these networks define “objectness”?

“Selective Search for object recognition”

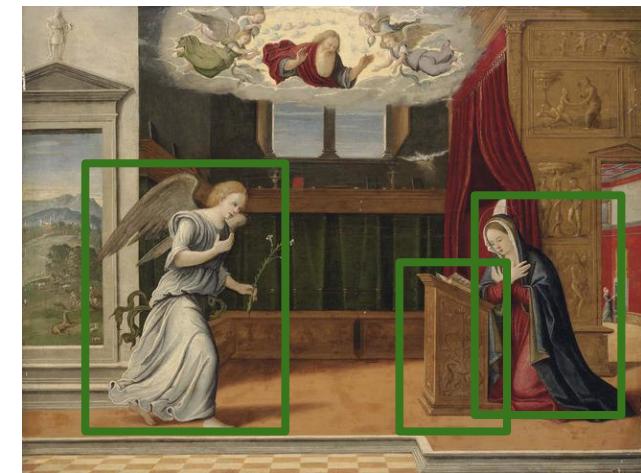
Object recognition

Find objects and recognize them



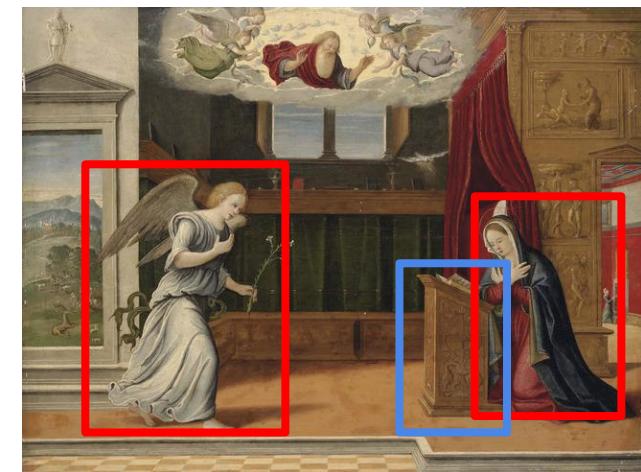
Object recognition

Find objects and recognize them



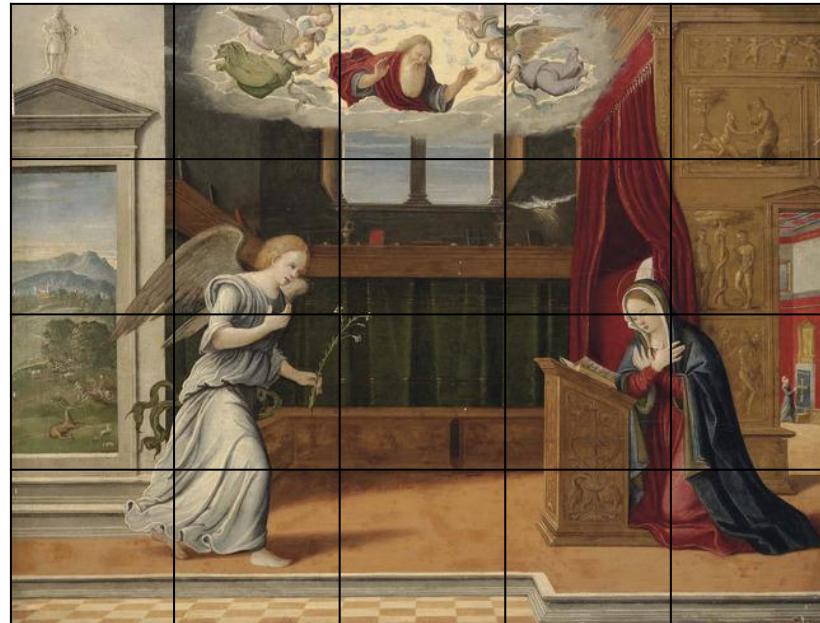
Object recognition

Find objects and *recognize* them



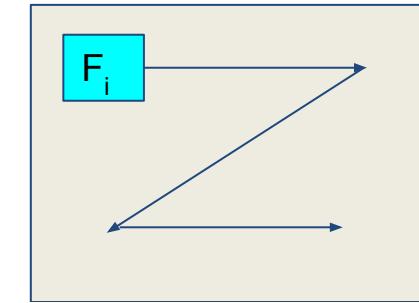
Person
Book shelf

Exhaustive search



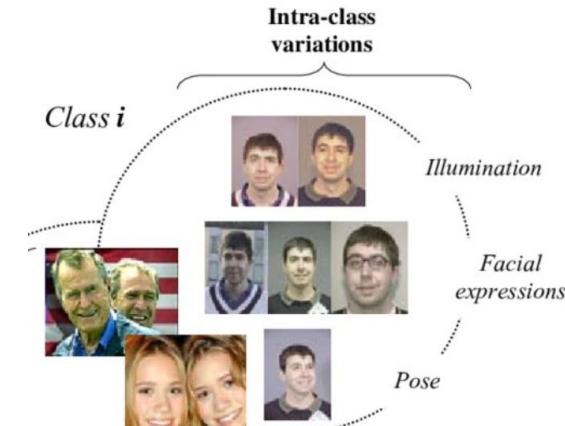
Sliding Window Object Detector

- Localizing → Identifying objects
- F_i : fixed size windows
- $f(F_i) \rightarrow ?$ (which class), f : classifier



Issues:

- Lacks contextual cues
- Fails when intra-class variations



Computationally effective?

- + Trained classification network can be used for object detection directly
- Computationally inefficient: One pass through the network for every patch!

Solution - Improve speed by using ***Fully Convolutional Kernels***

Still we would need ***regions of interest*** right and that too in an automated fashion maybe?

Selective Search

- Goals:
 - Capture all scales - How could we know the size of object?
 - Diversifications - Different criteria for segmentation
 - Fast to compute
- use of the powerful ***Bag-of-Words model for recognition*** [Further Reading]

Approach

- Hierarchical segmentation
 - Apply existing algorithms to find sub-segmentations
 - Small segmentations
 - Recursively combine small segmentations into big segmentations
 - Big segmentations



Algorithm

Algorithm 1: Hierarchical Grouping Algorithm

Input: (colour) image

Output: Set of object location hypotheses L

Obtain initial regions $R = \{r_1, \dots, r_n\}$ using [13]

Initialise similarity set $S = \emptyset$

foreach Neighbouring region pair (r_i, r_j) **do**

Calculate similarity $s(r_i, r_j)$

$S = S \cup s(r_i, r_j)$

while $S \neq \emptyset$ **do**

Get highest similarity $s(r_i, r_j) = \max(S)$

Merge corresponding regions $r_t = r_i \cup r_j$

Remove similarities regarding $r_i : S = S \setminus s(r_i, r_*)$

Remove similarities regarding $r_j : S = S \setminus s(r_*, r_j)$

Calculate similarity set S_t between r_t and its neighbours

$S = S \cup S_t$

$R = R \cup r_t$

Extract object location boxes L from all regions in R

**Felzenszwab
paper**

Algorithm

Algorithm 1: Hierarchical Grouping Algorithm

Input: (colour) image

Output: Set of object location hypotheses L

Obtain initial regions $R = \{r_1, \dots, r_n\}$ using [13]

Initialise similarity set $S = \emptyset$

foreach Neighbouring region pair (r_i, r_j) **do**

Calculate similarity $s(r_i, r_j)$

$S = S \cup s(r_i, r_j)$

while $S \neq \emptyset$ **do**

Get highest similarity $s(r_i, r_j) = \max(S)$

Merge corresponding regions $r_t = r_i \cup r_j$

Remove similarities regarding $r_i : S = S \setminus s(r_i, r_*)$

Remove similarities regarding $r_j : S = S \setminus s(r_*, r_j)$

Calculate similarity set S_t between r_t and its neighbours

$S = S \cup S_t$

$R = R \cup r_t$

Extract object location boxes L from all regions in R

Diversification

- How to model similarity?
- Depending on the input domain -> combination of different criteria
- What *criteria* for combining the segmentations?

Diversification

- Complementary Color Spaces

Diversification

- Complementary Color Spaces
- Complementary Starting Regions

Diversification

- Complementary Color Spaces
- Complementary Starting Regions
- Complementary Similarity Measures
 - a. Color similarity
 - b. Texture similarity
 - c. Size similarity
 - d. Shape compatibility - shape fill

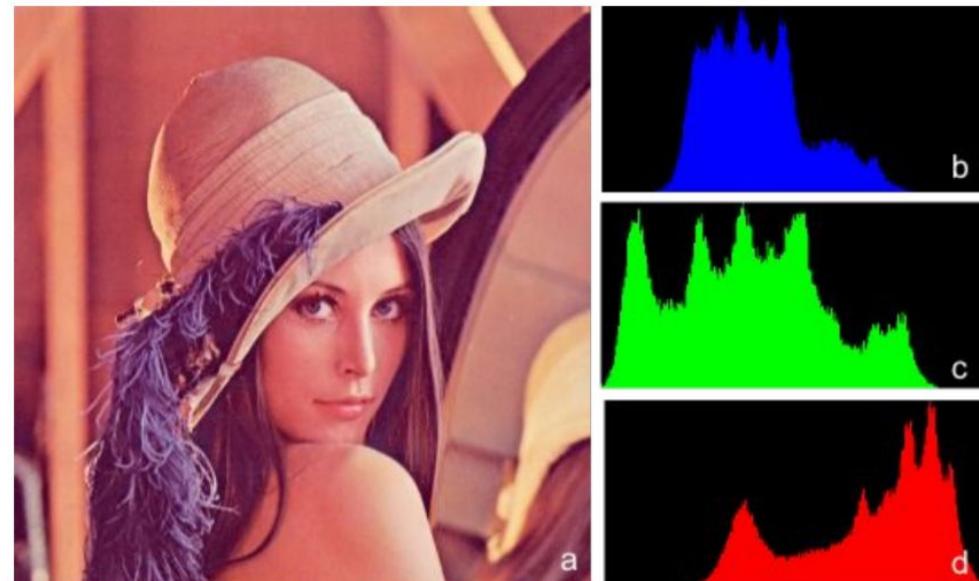
Color similarity

$$C_i = \{c_i^1, \dots, c_i^n\}$$

$$s_{colour}(r_i, r_j) = \sum_{k=1}^n \min(c_i^k, c_j^k)$$

$$C_t = \frac{\text{size}(r_i) \times C_i + \text{size}(r_j) \times C_j}{\text{size}(r_i) + \text{size}(r_j)}$$

$$\text{size}(r_t) = \text{size}(r_i) + \text{size}(r_j)$$



Texture similarity

- Extract derivatives in 8 directions for 3 channels
- 10 bins for each, 240 bins in total
- L1 Normalize
- Histogram Intersection

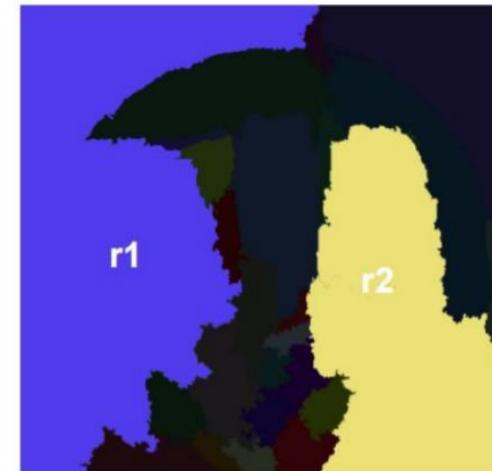
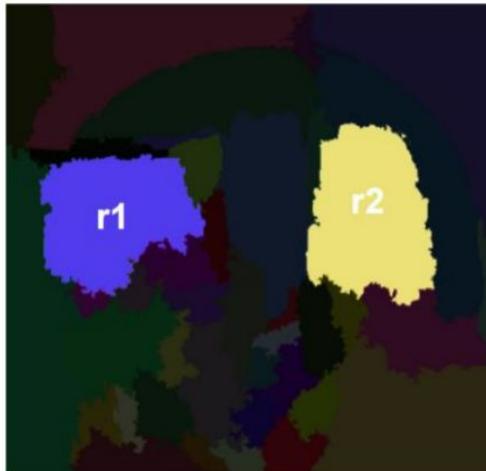
$$T_i = \{t_i^1, \dots, t_i^n\}$$

$$s_{texture}(r_i, r_j) = \sum_{k=1}^n \min(t_i^k, t_j^k)$$

Size similarity

- Prioritize **merging two small regions** into a large segmentation

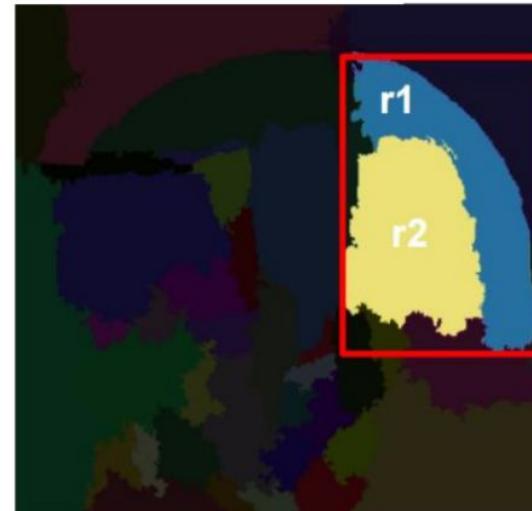
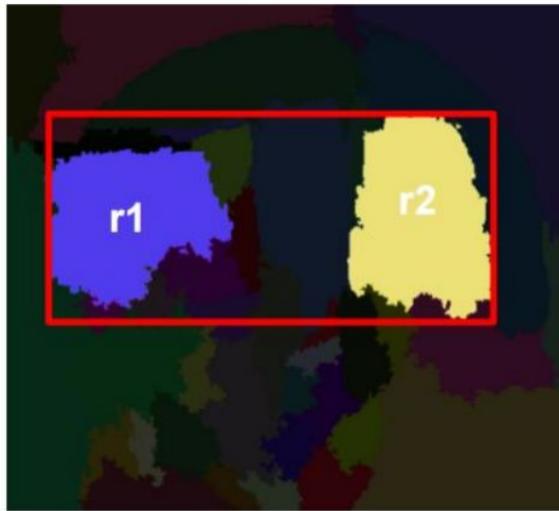
$$s_{size}(r_i, r_j) = 1 - \frac{\text{size}(r_i) + \text{size}(r_j)}{\text{size}(im)}$$



Shape compatibility

- Whether two segmentations fit each other?

$$fill(r_i, r_j) = 1 - \frac{\text{size}(BB_{ij}) - \text{size}(r_i) - \text{size}(r_j)}{\text{size}(im)}$$



Bounding
Box

Diversification (contd.)

- Weighted mixture approach

$$s(r_i, r_j) = a_1 s_{colour}(r_i, r_j) + a_2 s_{texture}(r_i, r_j) + \\ a_3 s_{size}(r_i, r_j) + a_4 s_{fill}(r_i, r_j),$$

Exercise Selective Search

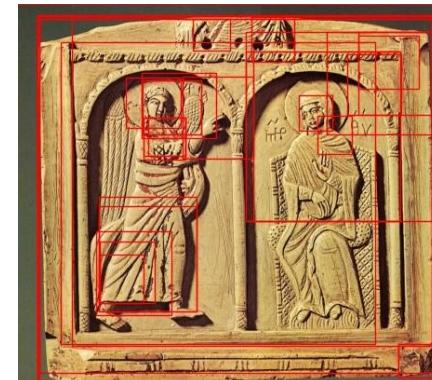
What you get?

- 9 images (3 from each of Art History, Class. Arch and Chris. Arch)

Your goal:

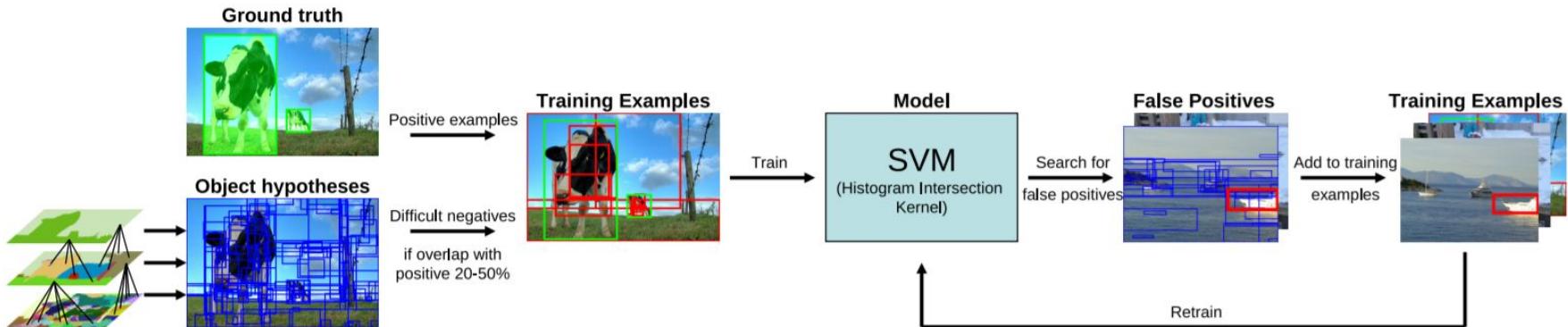
- Implement selective search
- Generate possible boxes that contains objects

Detailed description in exercise.pdf



Object Recognition

Approach : Selective Search + SIFT + SVM



Bonus Exercise I

- Implement Object Recognition Pipeline
- Use box proposals
- Extract Features from box proposals
- Train SVM
- Plug it all together

Evaluation

- Average Best Overlap (ABO)

$$\text{ABO} = \frac{1}{|G^c|} \sum_{g_i^c \in G^c} \max_{l_j \in L} \text{Overlap}(g_i^c, l_j)$$

Overlap between ground truth
and best selected box.

Average of “best overlaps” across all images.

$$\text{Overlap}(g_i^c, l_j) = \frac{\text{area}(g_i^c) \cap \text{area}(l_j)}{\text{area}(g_i^c) \cup \text{area}(l_j)}$$

Evaluation

Similarities	MABO	# box	Colours	MABO	# box
C	0.635	356	HSV	0.693	463
T	0.581	303	I	0.670	399
S	0.640	466	RGB	0.676	395
F	0.634	449	rgI	0.693	362
C+T	0.635	346	Lab	0.690	328
C+S	0.660	383	H	0.644	322
C+F	0.660	389	rgb	0.647	207
T+S	0.650	406	C	0.615	125
T+F	0.638	400	Thresholds		MABO
S+F	0.638	449	50	0.676	395
C+T+S	0.662	377	100	0.671	239
C+T+F	0.659	381	150	0.668	168
C+S+F	0.674	401	250	0.647	102
T+S+F	0.655	427	500	0.585	46
C+T+S+F	0.676	395	1000	0.477	19

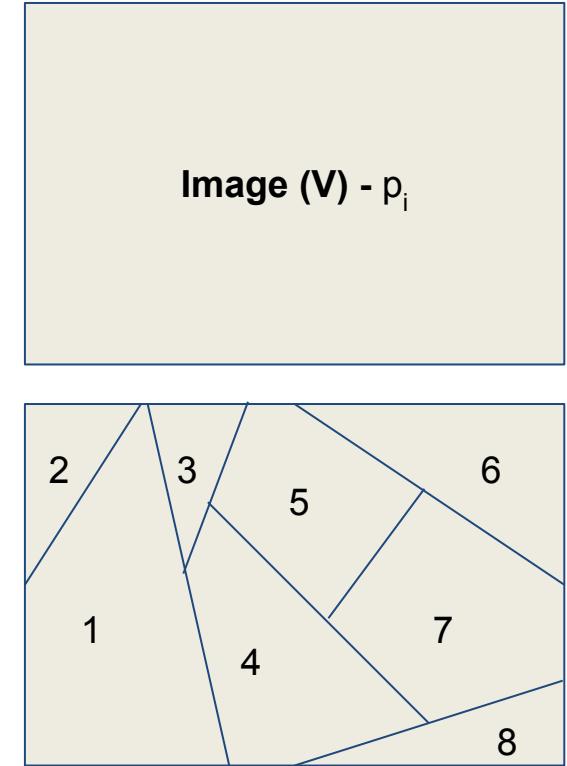


Evaluation

Version	Diversification Strategies	MABO	# win	# strategies	time (s)
Single Strategy	HSV C+T+S+F $k = 100$	0.693	362	1	0.71
Selective Search Fast	HSV, Lab C+T+S+F, T+S+F $k = 50, 100$	0.799	2147	8	3.79
Selective Search Quality	HSV, Lab, rgI, H, I C+T+S+F, T+S+F, F, S $k = 50, 100, 150, 300$	0.878	10,108	80	17.15

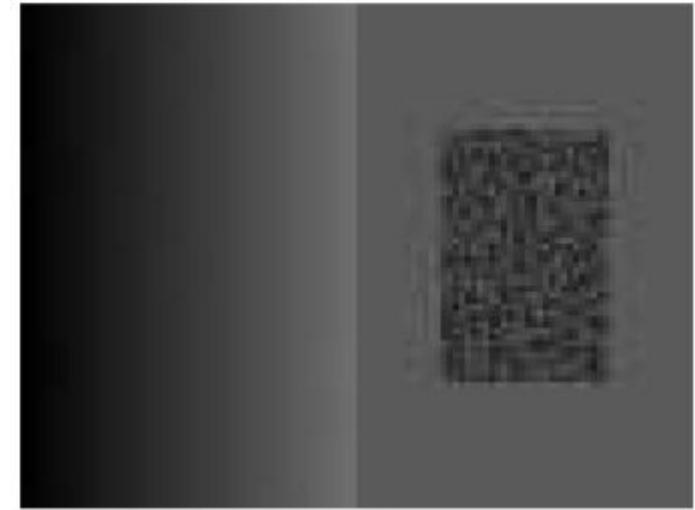
Felzenszwab - in a nutshell

- Divide an image into regions
- Defining boundaries by graph-representation satisfying global properties
- Perceptual grouping - psychology
- Idea
 - Capture perceptually important groupings
 - Time efficient - real time applications
- Method - **Pairwise Region Comparison Predicate**



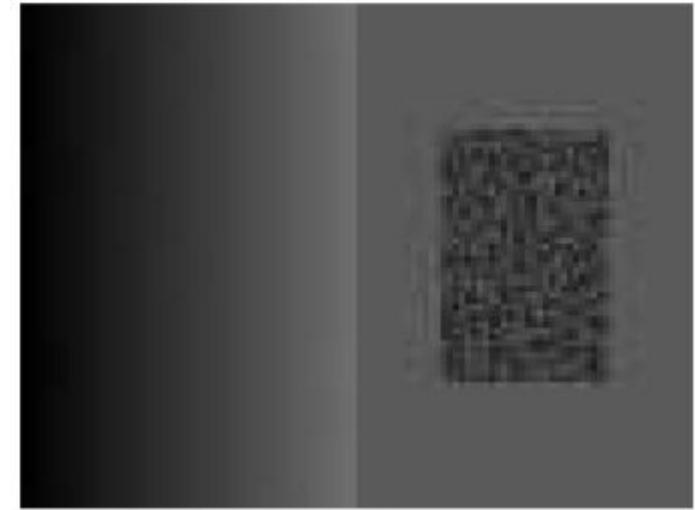
Understanding segmentation

- How many regions?



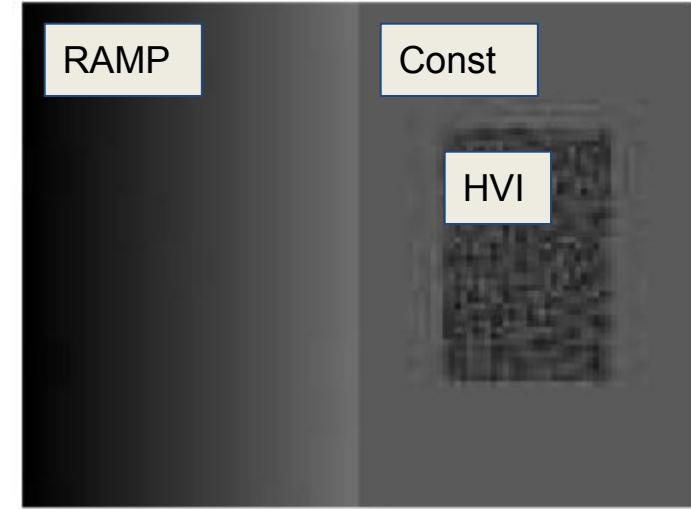
Understanding segmentation

- How many regions?
- Assumptions
 - Widely varying intensities
 - Purely local decision ***criteria***



Understanding segmentation

- How many regions?
- Assumptions
 - Widely varying intensities
 - Purely local decision criteria
- What we need?
 - Adaptive / non-local criterion



Pairwise Region Comparison Predicate

- Compares inter-component difference to within component differences
- Or formally:

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases}$$

Inter-Class Difference

- Minimum difference on the border between two components
- Or formally:

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j)) .$$

Intra-Class Difference

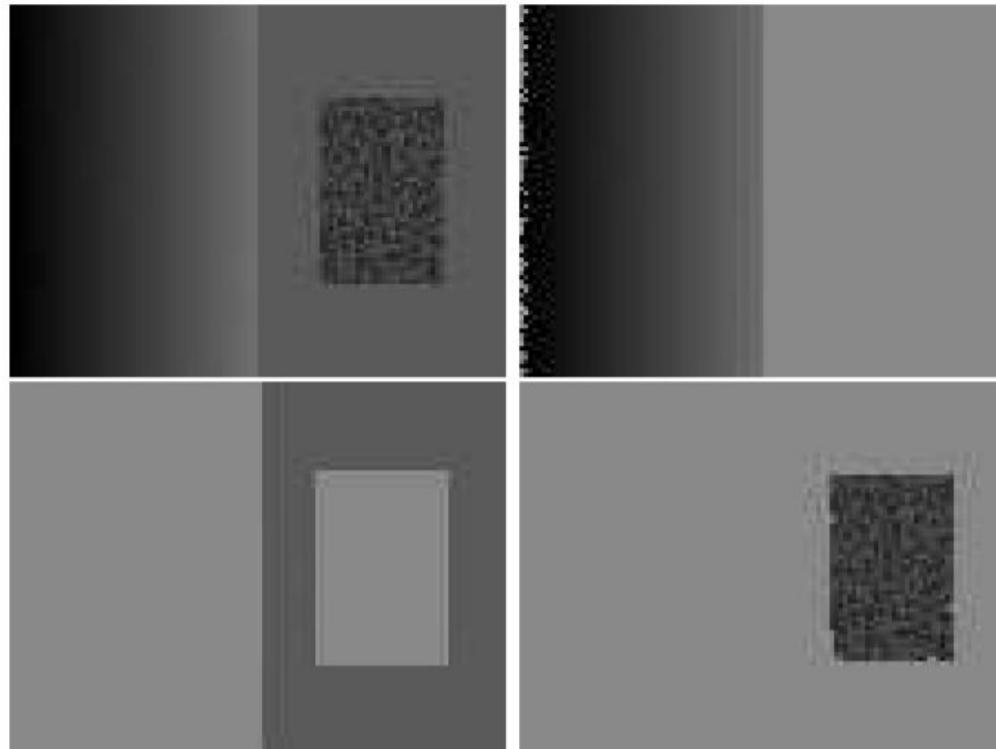
- Weighted maximum of internal differences
- Require stronger evidence for boundary for small components
- Or formally:

$$MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)).$$

$$Int(C) = \max_{e \in MST(C,E)} w(e) .$$

$$\tau(C) = k/|C|$$

Felzenszwab - Results I



Felzenszwab - Results II

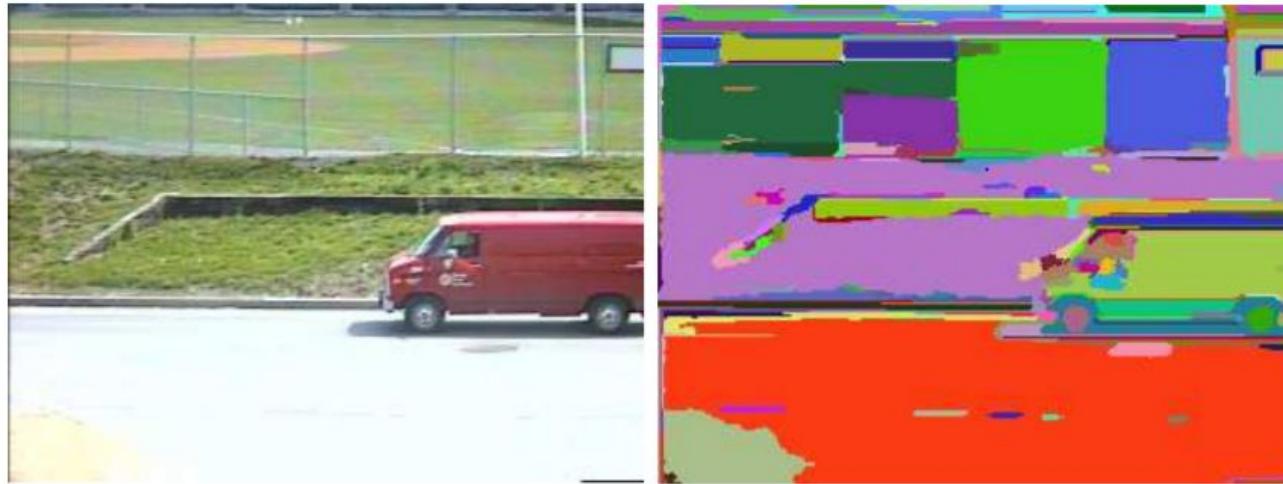
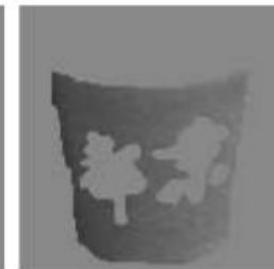
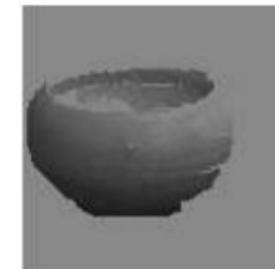
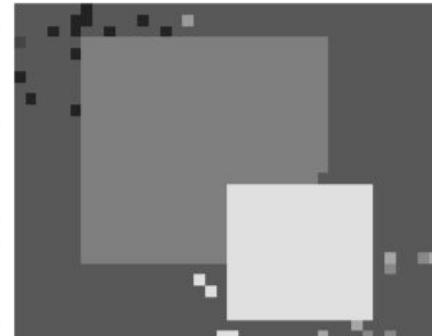
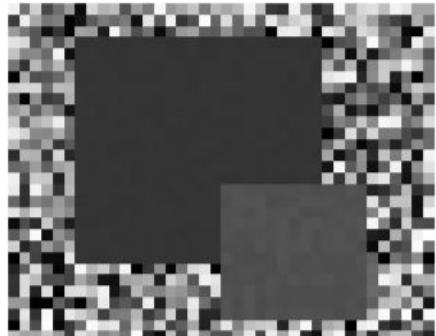


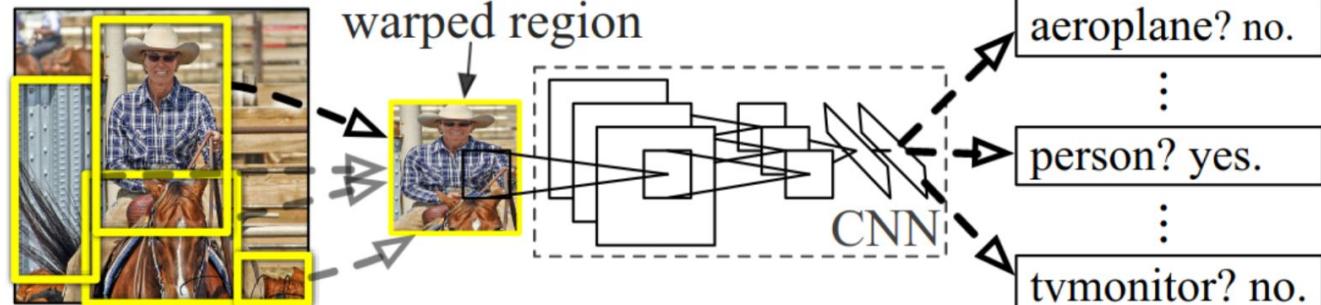
Figure 2: A street scene (320×240 color image), and the segmentation results produced by our algorithm ($\sigma = 0.8$, $k = 300$).

Felzenszwab - Result III



From Selective Search to Deep Learning: R-CNN [1]

R-CNN: *Regions with CNN features*



1. Input image

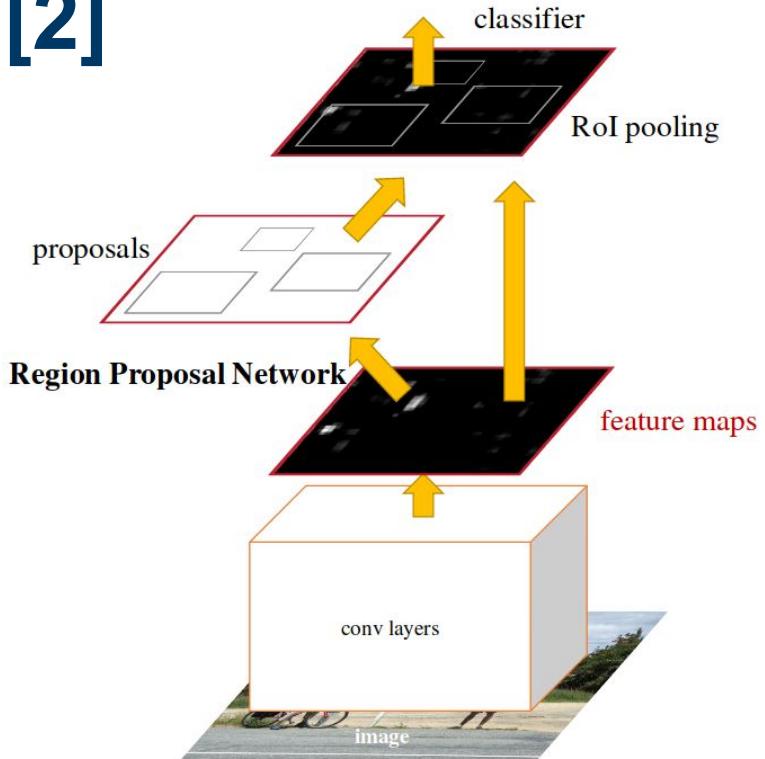
2. Extract region proposals (~2k)

3. Compute CNN features

4. Classify regions

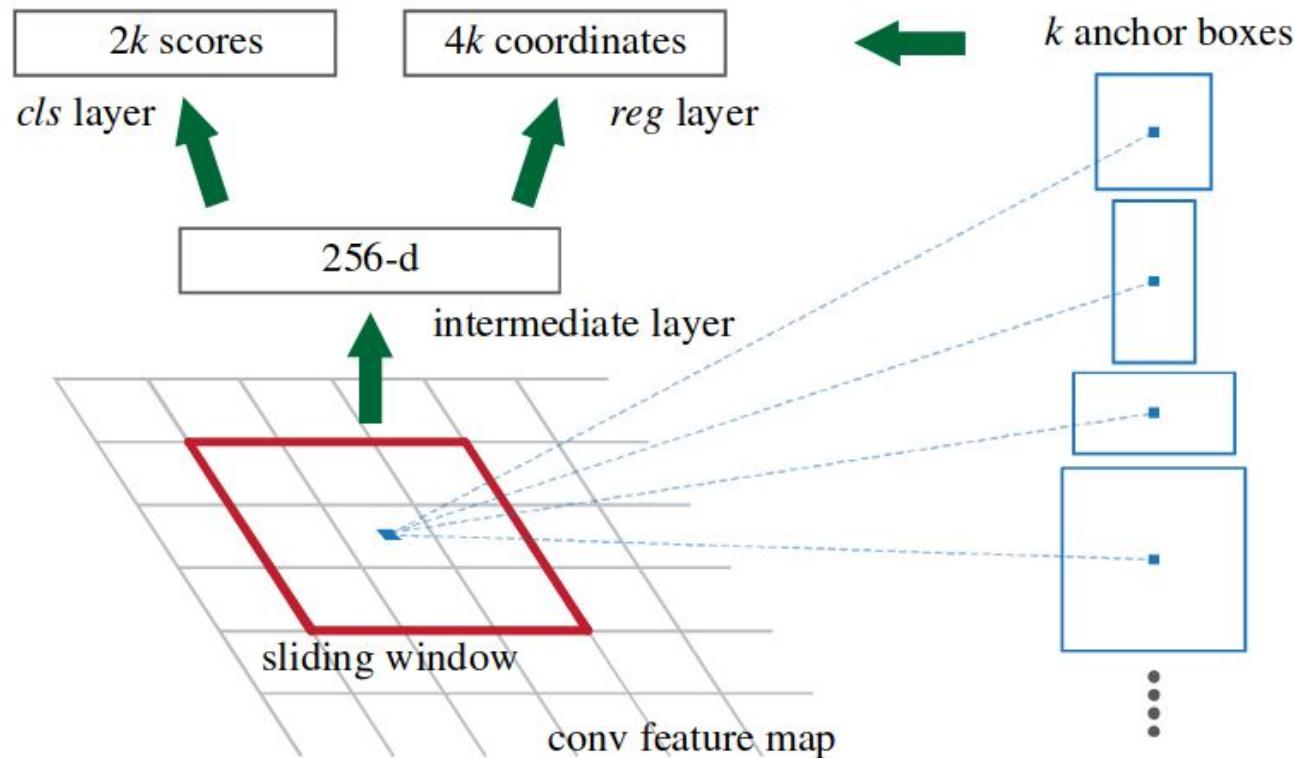
[1] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.

From Selective Search to Deep Learning: RPNs [2]

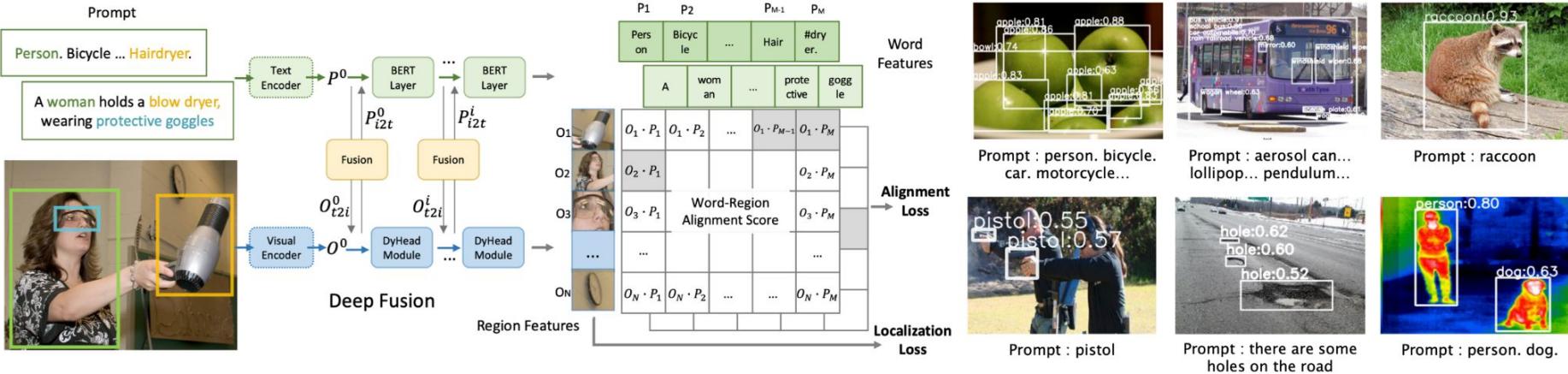


[2] Ren et al. Towards Real-Time Object Detection with Region Proposal Networks

From Selective Search to Deep Learning: RPNs



From Selective Search to Deep Learning: Multimodal Detection [3]



[3] Li, Liunian Harold, et al. "Grounded language-image pre-training." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022.

Further Reading

1. Felzenszwalb paper -
http://cs.brown.edu/people/pfelzens/papers/seq_ijcv.pdf
2. Selective Search paper -
<http://huppelen.nl/publications/selectiveSearchDraft.pdf>
3. Project on bag of words:
<https://cs.brown.edu/courses/csci1430/proj3/>

Backup slides

GRAPH - based segmentation

- $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ - undirected graph
 - $v_i \in \mathbf{V}$ - vertices (set of elements to be segmented, i.e. *pixels*)
 - $(v_i, v_j) \in \mathbf{E}$ - edges (pairs of neighboring vertices)
 - $w(v_i, v_j)$ - non negative dissimilarity
 - Based on intensity difference, color, motion etc.
- **S : Segmentation** (partition of \mathbf{V} into components or region)
 - $C \in \mathbf{S}$ - connected component in a graph $\mathbf{G}' = (\mathbf{V}, \mathbf{E}')$, where $\mathbf{E}' \subseteq \mathbf{E}$

Intuition:

- Any segmentation induced by subset of its edges in E
- Edges between 2 vertices in same component/region → low weight

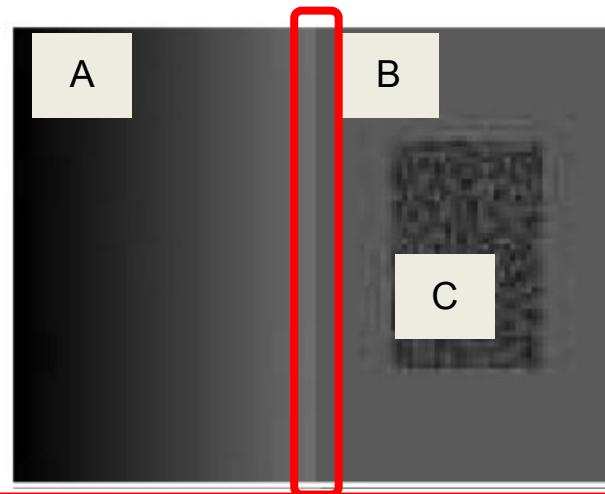
Pairwise Region Comparison Predicate

- **D**: predicate to evaluate whether or not there is an evidence for a boundary between 2 regions

“Measuring dissimilarity *between elements along the boundary of 2 components* relative to a measure of the dissimilarity *among neighboring elements within each of 2 components*”

Pairwise Region Comparison Predicate (PRCP)

- D: predicate to evaluate whether or not there is an evidence for a boundary between 2 regions



“Measuring dissimilarity *between elements along the boundary of 2 components* relative to a measure of the dissimilarity *among neighboring elements within each of 2 components*”

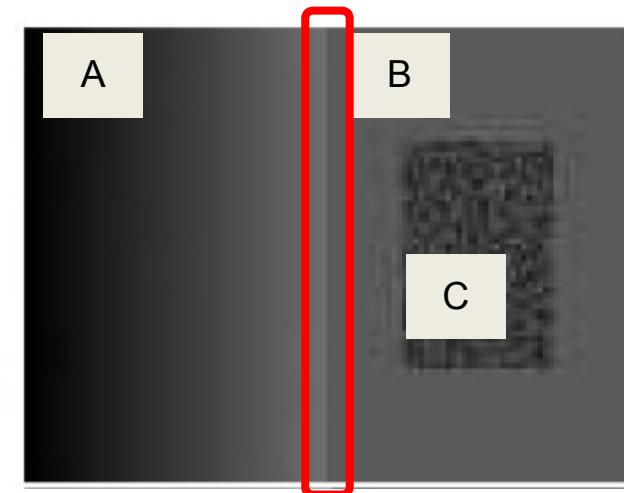
Pairwise Region Comparison Predicate (PRCP)

- Internal difference of a component

$$Int(C) = \max_{e \in MST(C, E)} w(e)$$

- Difference between 2 components

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j))$$



Pairwise Region Comparison Predicate (PRCP)

- PRCP

$$D(C_1, C_2) = \begin{cases} \text{true} & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{false} & \text{otherwise} \end{cases}$$

where,

$$MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2))$$

Algorithm

Input: $\mathbf{G(V, E)}$ with n vertices and m edges

Output: $\mathbf{S} = (C_1, \dots, C_r)$ i.e. segmentation of \mathbf{V}

1. Sort \mathbf{E} into $\pi = (o_1, o_2, \dots, o_m)$ by non-decreasing edge weights

Algorithm

Input: $\mathbf{G(V, E)}$ with n vertices and m edges

Output: $\mathbf{S} = (C_1, \dots, C_r)$ i.e. segmentation of \mathbf{V}

1. Sort \mathbf{E} into $\pi = (o_1, o_2, \dots, o_m)$ by non-decreasing edge weights
2. Start with seg. \mathbf{S}_0 where v_i is in its own component

Algorithm

Input: $G(V, E)$ with n vertices and m edges

Output: $S = (C_1, \dots, C_r)$ i.e. segmentation of V

1. Sort E into $\pi = (o_1, o_2, \dots, o_m)$ by non-decreasing edge weights
2. Start with seg. S_0 where v_i is in its own component
3. Repeat Step 4 for $q = 1, 2, \dots, m$
4. Construct S_q from S_{q-1} as follows:
 - a. v_i and $v_j \rightarrow$ connected by qth edge $\rightarrow o_q = (v_i, v_j)$

Algorithm

Input: $\mathbf{G(V, E)}$ with n vertices and m edges

Output: $\mathbf{S} = (C_1, \dots, C_r)$ i.e. segmentation of \mathbf{V}

1. Sort \mathbf{E} into $\pi = (o_1, o_2, \dots, o_m)$ by non-decreasing edge weights
2. Start with seg. \mathbf{S}_0 where v_i is in its own component
3. Repeat Step 4 for $q = 1, 2, \dots, m$
4. Construct \mathbf{S}_q from \mathbf{S}_{q-1} as follows:
 - a. v_i and $v_j \rightarrow$ connected by qth edge $\rightarrow o_q = (v_i, v_j)$
 - b. If v_i and v_j are in disjoint components of \mathbf{S}_{q-1} and $w(o_q)$ is small compared to internal diff. Of both these components then merge these two components, else do nothing

Algorithm

Input: $\mathbf{G(V, E)}$ with n vertices and m edges

Output: $\mathbf{S} = (C_1, \dots, C_r)$ i.e. segmentation of \mathbf{V}

1. Sort \mathbf{E} into $\pi = (o_1, o_2, \dots, o_m)$ by non-decreasing edge weights
2. Start with seg. \mathbf{S}_0 where v_i is in its own component
3. Repeat Step 4 for $q = 1, 2, \dots, m$
4. Construct \mathbf{S}_q from \mathbf{S}_{q-1} as follows:
 - a. v_i and $v_j \rightarrow$ connected by qth edge $\rightarrow o_q = (v_i, v_j)$
 - b. If v_i and v_j are in disjoint components of \mathbf{S}_{q-1} and $w(o_q)$ is small compared to internal diff. Of both these components then merge these two components, else do nothing
5. Return $\mathbf{S} = \mathbf{S}_m$