On the relevance of edge-conditioned convolution for GNN-based semantic image segmentation using spatial relationships

Patty Coupeau [1,2] Jean-Baptiste Fasquel [1]

[1] University of Angers, LARIS - Systems Engineering Research laboratory

[2] PhD student at the University of Angers

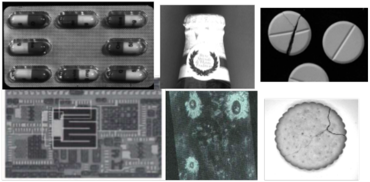
Summary

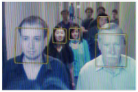
Summary

- Computer vision : many applications

Assembly & robotics

3D reconstruction





Augmented reality

123456789012

Traceability identification

Video-surveillance

- Computer vision : many situations

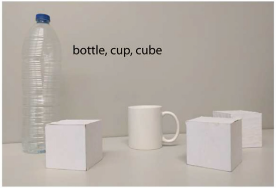
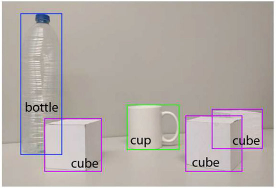
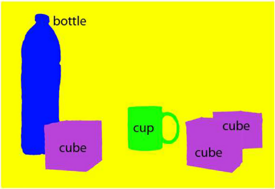


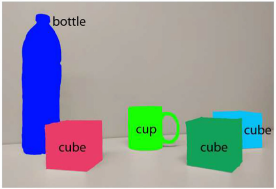
Image classification



Object localization



Semantic segmentation



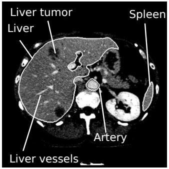
Instance segmentation

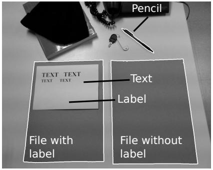
A. Garcia-Garcia, A survey on deep learning techniques for image and video semantic segmentation, Applied Soft Computing, 2018

Computer vision & structural information

- Often ignored : relationships between entities → structural information
  - Spatial, photometric, textural, geometric...
  - Motivation : a priori stability and simplicity of model declaration

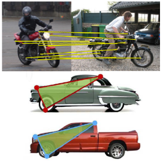
"Liver tumors" : "... included in..."
"Hypodense" : "... darker than..."





"Text written on " : "... included in..."
Similar brightness of files

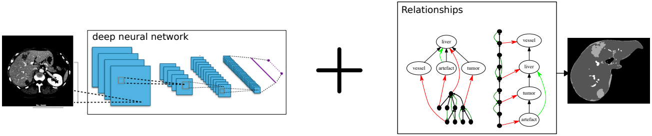
"... darker than..."
".... similarly bright...."
"Obvious" inclusion relationships



Geometric structures

O. Duchenne et al., IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011
J. Zhou et al., Journal of Visual Communication and Image Representation, 2015
J.B. Fasquel et al., IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019
I. Bloch, Fuzzy sets for image processing and understanding, Fuzzy Sets and Systems, 2015

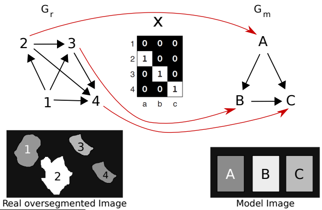
Preliminary semantic segmentation (e.g. CNN) + structural information = refined segmentation



How to exploit structural information ?

- Combinatorial optimization tools (e.g.constraint satisfaction problem, quadratic assignment problem)

Example : "On the right" + "Relative distances"



J. Chopin, J.B. Fasquel, H. Mouchere, R. Dahyot, and I. Bloch, 2020 10th International Conference on Image Processing Theory, Tools and Applications
J. Maciel and J.P.Costeira, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003
M. C. Vanegas, I. Bloch and J. Inglada, Fuzzy Sets and Systems, 2016

How to exploit structural information ?

- Graph neural network (GNN) : learn the matching (node classification)

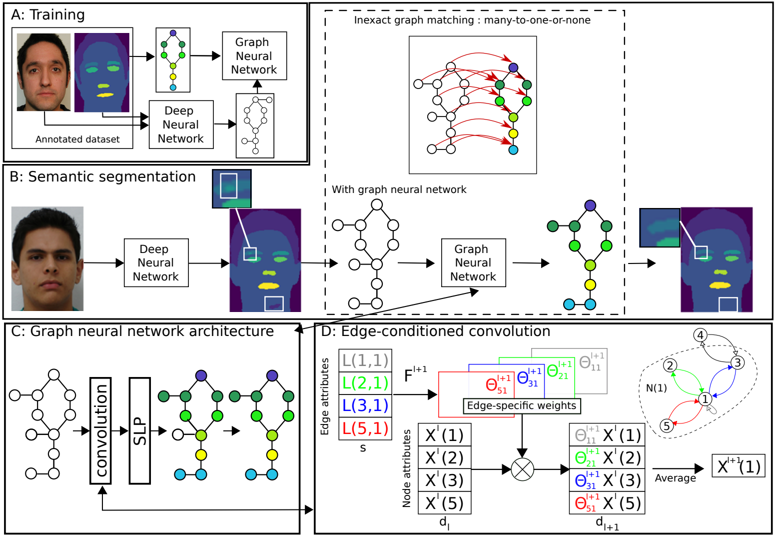
Constraints:

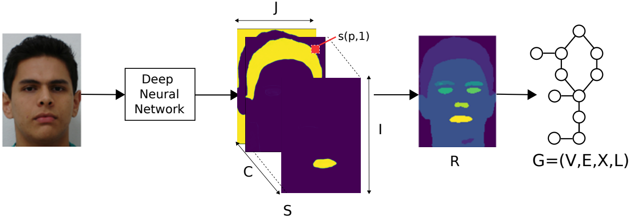
- Managing graphs of arbitrary size (depends on the CNN output)
- Managing both node and edge attributes

Z. Zhang et al., IEEE Transactions on Knowledge and Data Engineering, 2020
A. Zanfir et al., 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition
A. S. Nassar et al., Computer Vision - ECCV 2020 - 16th European Conference, 2020

Summary

Overview


A: Training
Annotated dataset
Graph Neural Network
Deep Neural Network
Inexact graph matching : many-to-one-or-none
B: Semantic segmentation
With graph neural network
Deep Neural Network
Graph Neural Network
C: Graph neural network architecture
convolution
SLP
D: Edge-conditioned convolution
Edge attributes
L(1,1)
L(2,1)
L(3,1)
L(5,1)
s
F^{l+1}
Edge-specific weights
N(1)
Node attributes
X^l(1)
X^l(2)
X^l(3)
X^l(5)
d_l
d_{l+1}
Average
X^{l+1}(1)



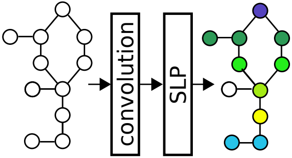
Segmentation map: $S \in R^{P \times C}$ from CNN

S(p, c) $\in [0, 1]$: probability of pixel p of belonging to class c

R: set of all resulting connected components

From R, construction of graph $G = (V, E, X, L)$

- V: set of nodes (each $v \in V$ corresponds to a region $R_v \in R$)
- E: set of edges
- $X : V \longrightarrow \mathbf{R}^C$: node attribute assignment function (average membership probability vector over the set of pixels $p \in R_v$)
- $L : E \longrightarrow \mathbf{R}^s$: edge attribute assignment function (depends on the considered spatial relationships)



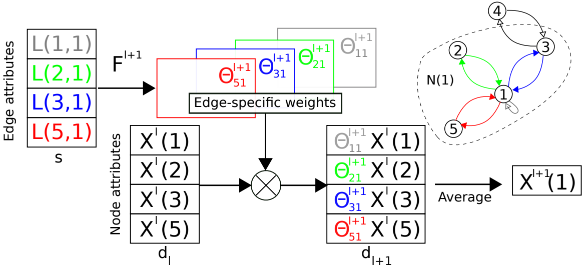
Node classification:

- arbitrary graph size
- attributes on nodes and edges

Only 2 layers:

- convolution: aggregating neighborhood information related to each node (message passing)
- single layer perceptron (SLP): $\mathbf{R}^{d^{l+1}} \longrightarrow \mathbf{R}^{C}$, providing a class membership probability vector to each node of the graph

D. Bacciu et al., Neural Networks, 2020

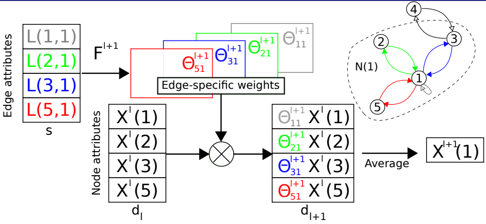


For node $i \in V$, ECConv computes a new attribute $X^{l+1}(i)$ by combining different information from layer l :

- the attributes of the set $N(i)$ of nodes ($N(i) = \{j|(j,i) \in E\} \cup \{i\}$)
- the attributes of the set of related edges $\{L(j,i)|j \in N(i)\}$

M. Simonovsky et al., 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)

Edge-conditioned convolution: ECConv



$$
\begin{aligned}
X^{l+1}(i) &= \frac{1}{|N(i)|} \sum_{j \in N(i)} F^{l+1}(L(j,i)) X^{l}(j) + b^{l+1} \\
&= \frac{1}{|N(i)|} \sum_{j \in N(i)} \Theta_{ji}^{l+1} X^{l}(j) + b^{l+1}
\end{aligned} \tag{1}
$$

$F^{l+1} : \mathbf{R}^{s} \longrightarrow \mathbf{R}^{d^{l+1} \times d^{l}}$ mapping function (a multi-layer perceptron in our case)

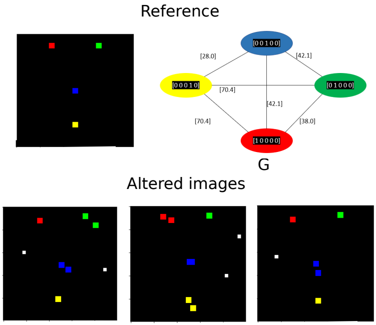
X^{l+1} is computed using the average operator (permutation invariant operator)

Dimensions of node attributes d^{l} ($l > 0$) are hyperparameters

Several convolution layers could be cascaded (only one in this study)

Summary

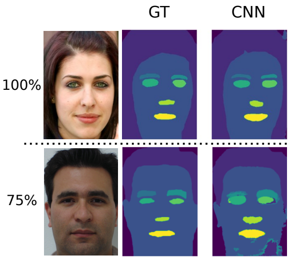
Synthetic



4 classes + background

100 altered images

FASSEG-Instances



8 classes + background

70 human faces

CNN: U-Net (splitting: 20/10/40)

Influence of the dataset size (100% / 75%)

---

https://github.com/Jeremy-Chopin/FASSEG-instances

J. Chopin et al., 2020 10th International Conference on Image Processing Theory, Tools and Applications (IPTA)

O. Ronneberger et al., Medical Image Computing and Computer-Assisted Intervention, 2015

Synthetic

- Node attributes: membership probability vector of the region R_i
- Edge attributes: distance between barycenters of the connected regions R_i and R_j $(L(i,j) = |b_i - b_j|)$

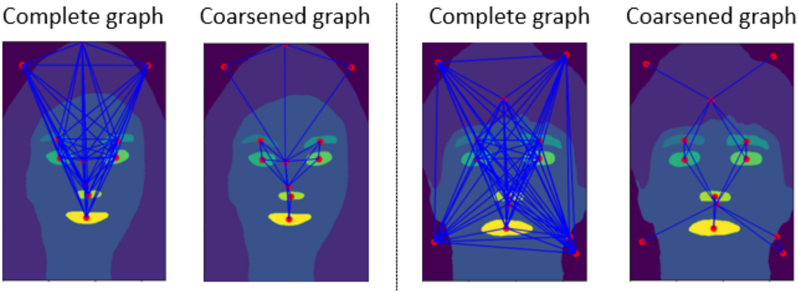
FASSEG

- Extraction of large connected components ($\leq$ 30 pixels): association to a node
- Node attributes: membership probability vector of the region R_i
- Edge attributes: minimum and maximum distance between the connected regions R_i and R_j $(L(i,j) = [d_{min}^{R_i,R_j}, d_{max}^{R_i,R_j}])$

Impact of the size of the neighborhood

Coarsened graph based on edge properties $L(i,j)$ $G_c = (V, E_c, X, L)$, where $E_c \subseteq E$

Hyperparameter radius ρ: limit distance between regions



D. Bacciu et al., Neural Networks, 2020

Table: Graphs parameters for synthetic dataset and FASSEG. Values indicated are a mean over all images of the test dataset. Number of classes (C), of nodes ($|V|$) and of edges ($|E|$ and $|E_c|$), where $|E_c|$ is the number of edges after coarsening

Dataset	C	$\|V\|$	$\|E\|$	$\|E_c\|$
Synthetic	5	7 (max: 14)	44 (max: 90)	9 (max: 12)
FASSEG 100%	9	12 (max: 26)	172 (max: 650)	33 (max: 134)
FASSEG 75%	9	17 (max: 86)	378 (max: 3867)	99 (max: 728)

Table: Results of classification of synthetic data with different configurations of graphs and convolution operators.

Method	**Accuracy**
ECConv (G_c)	**1.00**
ECConv	0.98
GCNConv* (G_c)	0.59
ECConv (no node attributes)	0.20

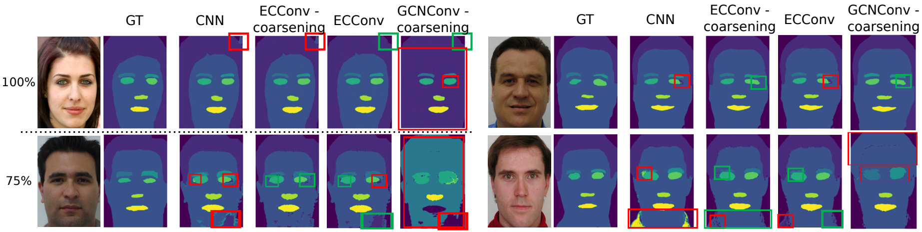
*GCNConv: does not consider edge attributes

T. Kipf et al., International Conference on Learning Representations, 2017

Results: FASSEG

Table: Segmentation results on FASSEG with CNN only and CNN followed by GNN (using ECConv or GCNConv). Complete graphs and coarsened ones are compared.

Method	75%			100%		
	DSC	***B-DSC***	***HD***	***DSC***	***B-DSC***	***HD***
CNN	0.798	0.675	54.40	0.845	0.745	27.20
ECConv	0.798	0.728	33.53	0.845	**0.769**	**19.76**
ECConv (G_c)	**0.804**	**0.731**	**32.00**	0.845	0.759	22.80
GCNConv (G_c)	0.537	0.470	124.87	0.599	0.516	100.95



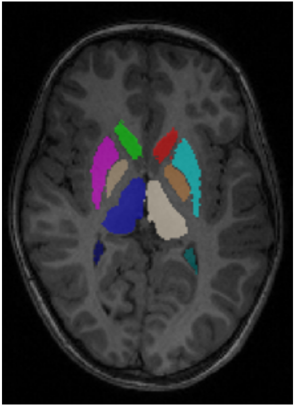
Summary

- GNN-based technique with inexact graph matching procedure: improve CNN-based image segmentation
- Consideration of both node (CNN output) and edge (spatial relationships) attributes with ECConv: promising preliminary results
- Simple architecture (CONV + SLP) faster than combinatorial approaches likes QAP (inference time $\leq$ 5s)
- Structural information and graph coarsening makes algorithms more robust to small dataset

Preliminary experiments to be improved (larger datasets, GNN-architecture, etc.)

P. Coupeau, J.-B. Fasquel, M. Dinomais, "On the relevance of edge-conditioned convolution for GNN-based semantic image segmentation using spatial relationships", International Conference on Image Processing Theory, Tools and Applications, 2022 (accepted)

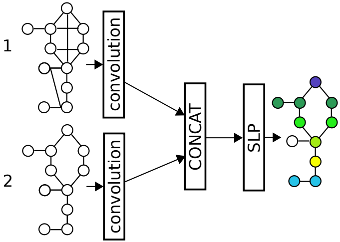
- Compare with more recent CNN-based method
- Larger dataset, applications more complex (medical images, etc.)



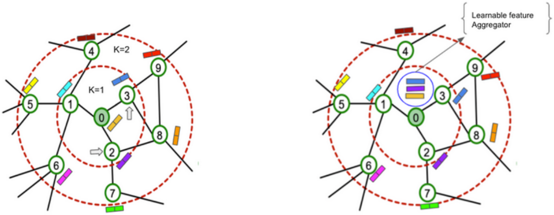
M. Lou et al., Neurocomputing, 2022
S. Chen et al., Medical Image Analysis, 2022
C. Oyarzun Laura et al., Methods, 2021

Benefit of coarsening: multi-coarsening ?
Parallel ?

$\rightarrow$ Hyperparameter ρ ?



Alternative: approach similar to GraphSAGE



Neighborhood Sampling of input graph at search depth K=1

Feature Aggregation for the target node **0** at K=1 with sampling

$\rightarrow$ to adapt to consider both node and edge attributes

W. Hamilton et al., in Advances in Neural Information Processing Systems, 2017

https://www.arangodb.com/2021/08/a-comprehensive-case-study-of-graphsage-using-pytorchgeometric/

Table: Segmentation results provided by the CNN only and our proposal. Results are provided for each class (not the background): Hr (hair), Fc (face), L-br (left eyebrow), R-br (right eyebrow), L-eye (left eye), R-eye (right eye), nose and mouth.

	75%						100%					
Method	**CNN**			**Proposal**			**CNN**			**Proposal**		
Class	***DSC***	***B-DSC***	***HD***	***DSC***	***B-DSC***	***HD***	***DSC***	***B-DSC***	***HD***	***DSC***	***B-DSC***	***HD***
Hr	0.924	0.773	126.26	**0.925**	**0.841**	**86.15**	0.941	0.825	85.18	0.941	**0.838**	**73.54**
Fc	0.948	0.917	48.29	**0.949**	**0.960**	**25.06**	0.957	0.955	24.38	0.956	**0.965**	**19.17**
L-br	0.681	0.547	65.33	**0.686**	**0.617**	**30.19**	0.751	0.679	11.41	0.751	0.678	11.41
R-br	0.667	0.537	65.77	0.652	**0.599**	**42.44**	0.744	0.584	42.50	0.745	**0.653**	**21.10**
L-eye	0.783	0.670	36.47	**0.804**	**0.707**	**23.06**	0.865	0.740	19.88	0.865	**0.782**	**10.11**
R-eye	0.783	0.643	36.97	0.783	**0.681**	**29.30**	0.837	0.718	14.29	0.837	**0.750**	**8.27**
Nose	0.742	0.559	41.41	**0.771**	**0.662**	**10.14**	0.797	0.684	8.47	0.797	**0.697**	**7.18**
Mouth	0.859	0.752	14.69	0.858	**0.779**	**9.42**	0.867	0.770	11.46	0.867	**0.791**	**7.31**

Nvidia Quadro RTX 3000 GPU - PyTorch libraries (torch_geometric.nn)

- optimizer: Adam
- loss function: negative log likelihood
- initial learning rate $lr_0 = 0.01$, reduction factor $\sigma = 5e-4$

Synthetic

- 250 epochs
- $d1$=6
- train: 70 / test: 30

FASSEG-Instances

- 600 epochs
- $d1$=7
- train: 30 / test: 40

https://pytorch-geometric.readthedocs.io/en/latest/modules/nn.html