Semantic segmentation for mapping hockey broadcast images in 2D plan

Philippe Blouin-Leclerc†, Stéphane Caron†

Department of Computer Science and Software Engineering, Université Laval

†Authors contributed equally to this work.

Introduction

We propose a novel way to recognize key locations within hockey broadcast images using semantic segmentation and convolutional neural networks (CNN). We implement a network that learn this semantic and could then be used for many applications such as mapping a broadcast image into a 2D plan.

Motivations :

- Computer vision allows the detection of many events at the same time, which is well suited for sports analytics data collection.
- Semantic segmentation is often a key step as it brings a **general understanding** of the image.

Related work :

- Homayounfar and al. (2017) : Sports field localization via deep structured models.
- Ronneberger and al. (2015) : Convolutional networks for biomedical image segmentation (U-Net).

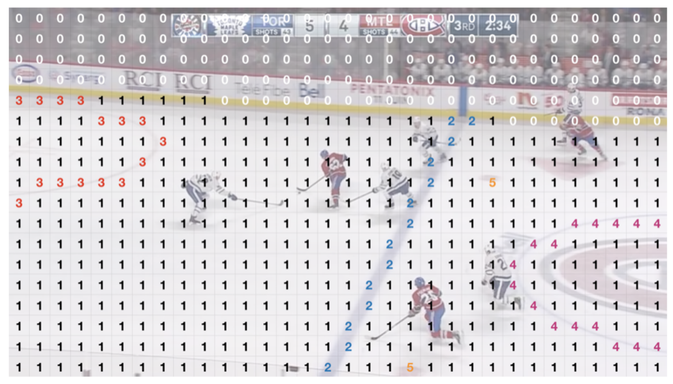
Goals :

- Evaluate the capability of CNN to learn the semantic representation of a hockey ring surface broadcast image.
- Provide meaningfull insights on how to build architectures that can learn well every components of an image.
- Propose a method that uses semantic segmentation representation to map objects and events into a 2D plan.

Semantic segmentation background

Semantic segmentation is a computer vision task where the model learns the general representation of an image by attributing a label to each and every pixels.

Define the task : In order to make pixel-wise predictions, we need to have a representation saying which class is attached to each label. This representation is what we call a **mask** (see right-side image below).



As many classification problems, we need to one-hot encode all labels (one matrix for each class) which mean we can summarize the dimensions workflow as follow for one 6 classes RBG image :

$$(NbChannels, Height, Width) \Rightarrow (NbLabels, Height, Width) \Rightarrow (1, Height, Width)$$
$$(3, 256, 451) \Rightarrow (6, 256, 451) \Rightarrow (1, 256, 451)$$

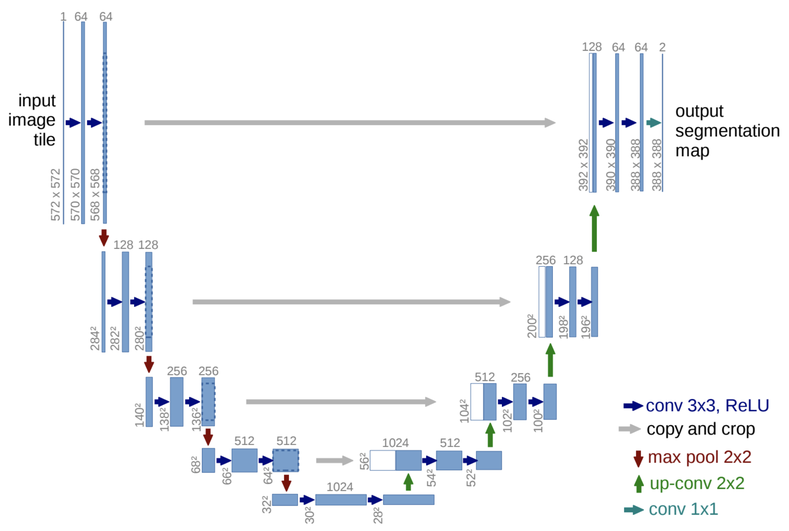
Methodology

Our methodology is splitted in 3 main components :

1. Set up
   - Dataset creation
     - XX NHL broadcast images
   - Labeling task : cvat tool
     - 9 classes : crowd, ice, blue line, red line, goal line, circle zones, middle circle, dots and boards)
2. Semantic segmentation
   - Architecture set up
   - Loss definition
   - Data augmentation
   - Training details
3. Mapping to 2D plan
   - Key points recognition
   - 2D translation

Architecture and training experiments

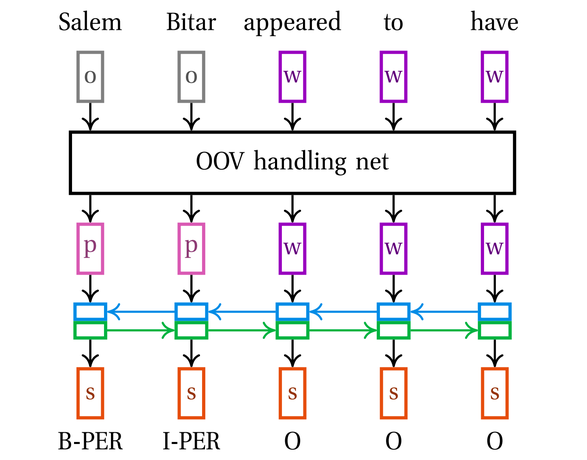
U-Net : To perform our segmentation, we chose an architecture called U-Net. This network is **fast** and can be trained with **few images**.



Loss definition :

Training details :

2D mapping method



Two nets working together : the first predicts OOV embeddings (see OOV handling net section) and the second one predicts tags.
The simple architecture of the labeling net is used to emphasize the usefulness of our module, and to minimize the influence of other factors.

Results

Task	Tag	Ex.	Ponderation		
			Word	Left	Right
NER	O	1039	**0.81**	0.08	0.11
	B-PERS	63	0.21	0.31	**0.49**
	I-PER	119	0.16	**0.52**	0.32
	B-ORG	40	0.26	0.30	**0.44**
	I-ORG	3	0.27	0.31	**0.42**
	B-LOC	13	0.23	0.30	**0.47**
	I-LOC	2	0.16	**0.48**	0.36
	B-MISC	47	**0.40**	0.21	0.39
	I-MISC	5	**0.41**	0.26	0.33
POS	NNP	308	0.29	0.31	**0.40**
	NN	46	**0.45**	0.20	0.35
	CD	827	**0.86**	0.05	0.09
	NNS	23	**0.37**	0.24	**0.39**
	JJ	100	**0.49**	0.15	0.36

Average weights assigned to word's characters, left context and right context by the attention mechanism. We can clearly see the shift of attention according to the target entity. We also observe that the attention depends on the task at hand.

Conclusion

Discussion :

- **Morphology** and **context** help predict useful embeddings.
- **The attention mechanism works** : depending on the task, the network will use either more the context or the morphology to generate an embedding.

Future works :

- Apply the **attention mechanism on each character of the OOV word and each word of the context** instead of using the hidden state of the respective elements only.
- Test our attention model in **different languages** and on other NLP tasks, such as **machine translation**.