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).

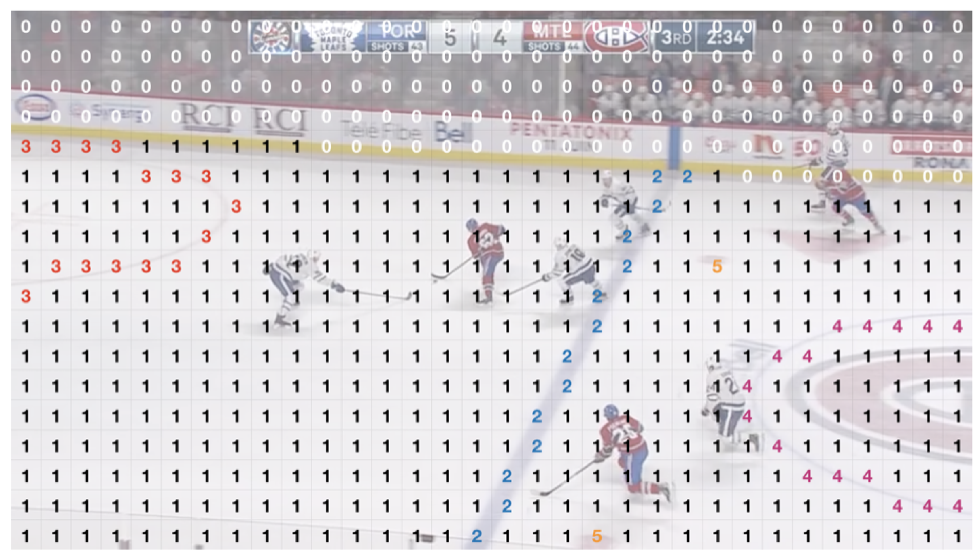
Goals :

- 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.

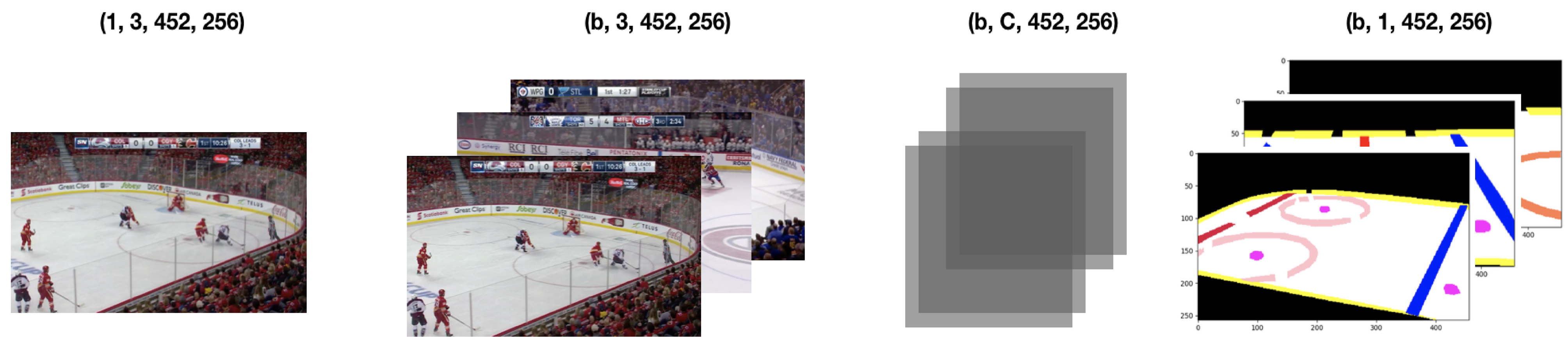
Semantic segmentation background

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).



We can summarize the workflow as follow where b is batch size and C is the number of classes :



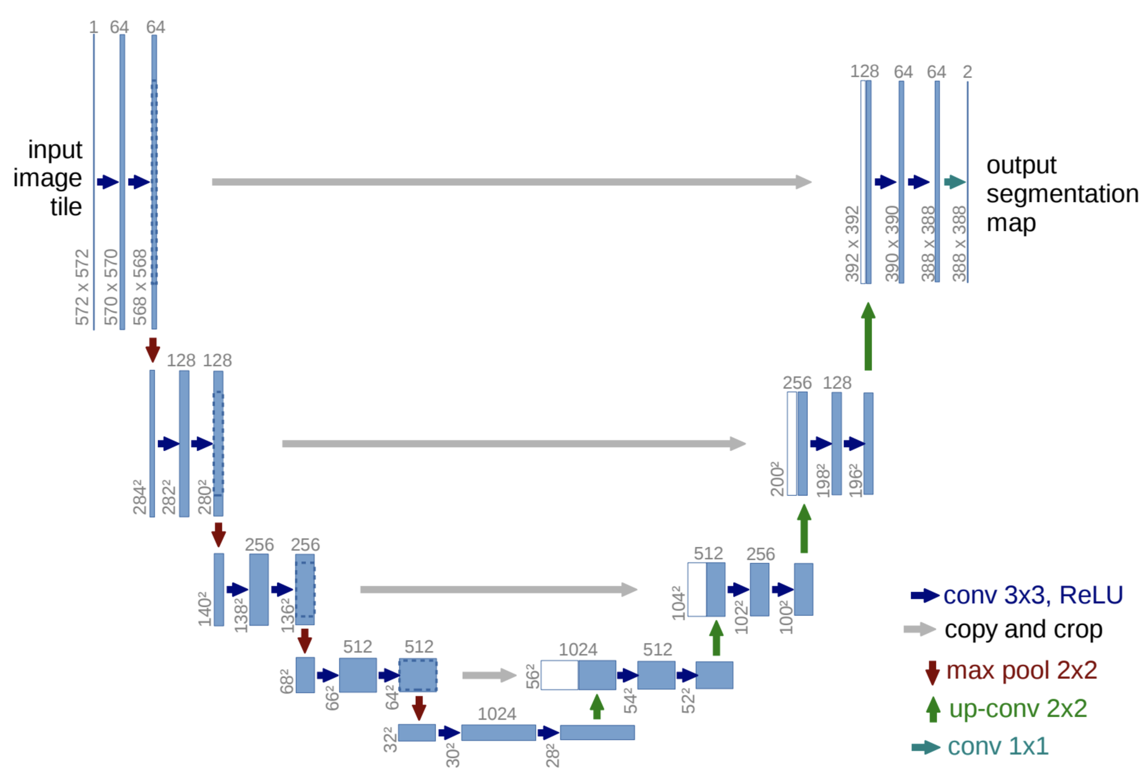
Methodology

Our methodology is splitted in 3 main components :

1. **Set up**
   - Dataset creation
     - 43 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 classes : crowd and ice
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**. No pre-trained network found for that model.



VGG16 : We also decided to use a **pre-trained** VGG16 architecture without the max-pooling steps. The resuting output we'll then the **same size** as the input.

Loss definition : We defined 2 kinds of loss :

- Cross-Entropy loss
- Dice loss

The 9-classes problem is suffering from **class imbalance** (there is much more pixels of ice/crowd than lines or dots. We address that problem in the dataset labeling and in the loss definition. For Cross-Entropy loss, we adapted the weights for loss depending on the label frequency. We also implemeted the Dice loss :

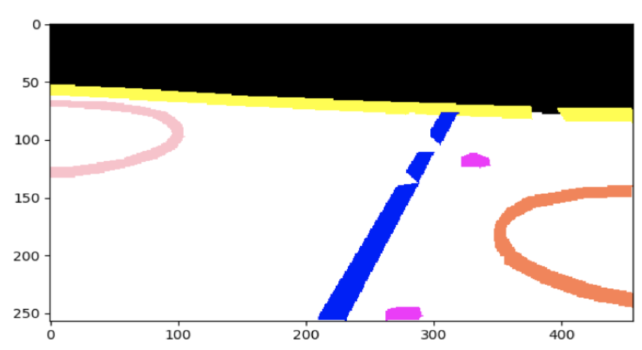
$$\text{Dice Loss} = \frac{1}{nb\_class} * \sum_{i=1}^{nb\_class} \left(1 - \frac{2\sum_{pixels} y_{true} y_{pred}}{\sum_{pixels} y_{true}^2 + \sum_{pixels} y_{pred}^2}\right)$$

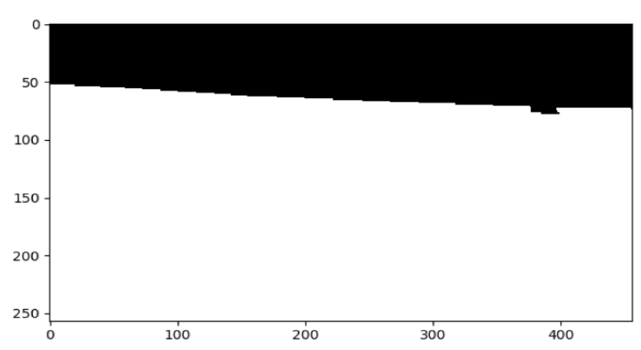
Training details :

- Different learning rates (trainning schedule) and batch size
- SGD optimizer with momentum and weight decay
- Different upsampling methods (bilinear and transpose convolutions)

Dataset

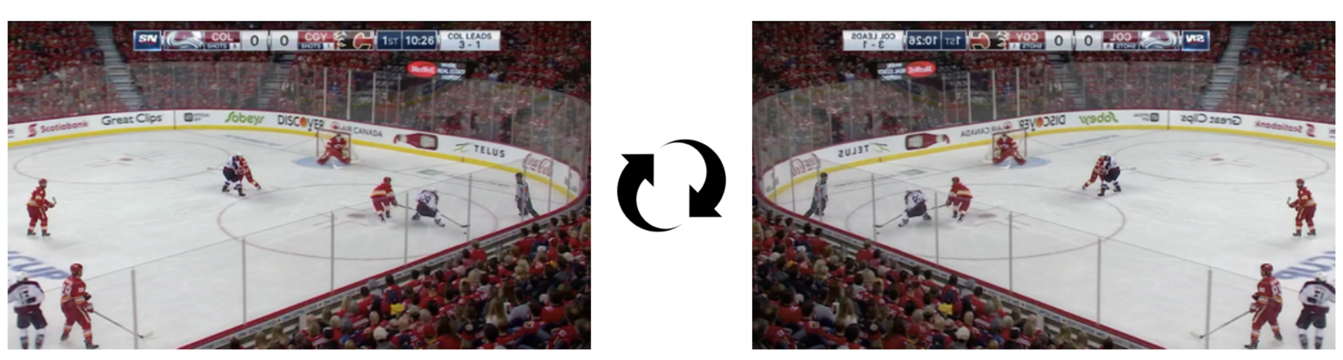
We created our own dataset by making screenshots of NHL broadcast games and labeled them. Here is an example of the after the labeling task (for both 9 classes (left) and 2 classes (right)) :





To adress class imbalance, we draw larger areas around rare labels pixels such as dots, circles and lines.

Because we only had a total 43 images, we augmented our train dataset by making **horizontal rotation**. That kind of transformation makes sense in the context of a hockey ring (symmetry).



Results

Here are the results we gathered from our best experiements :

Labels	Model	Epochs	Loss	Validation Loss	Test Loss
9-classes	U-Net	40	Dice	0.08	0.11
	VGG16	40	Dice	0.31	**0.49**
2-classes	U-Net	30	CE	0.31	**0.40**
	VGG16	20	CE	0.20	0.35

Here is 2 samples that shows our best results for both 9-classes and 2-classes models :

Mettre des exemples d'images de outputs avec les ground truth

Conclusion

Discussion :

- For 9-classes predictions, we need more images.
- For 2-classes predictions, bla bla bla
- Bla bla bla

Future works :

- Extract and label **more images** (was time consuming)
- Starts from a pre-trained model as our encoder and build a **proper decoder** for this specific task.
- Use the semantic segmentation learned by the model to map key areas on the ice into a 2D plan.