# Pix2Seq

This paper advocates a new approach, based on the intuition that if a neural net knows about where and what the objects are, we just need to teach it to read them out. And by learning to “describe” objects the model can learn to ground the “language” on pixel observations, leading to useful object representations.

Given an image, our model produces a sequence of discrete tokens that correspond to object descriptions (e.g., object bounding boxes and class labels), reminiscent of an image captioning system.

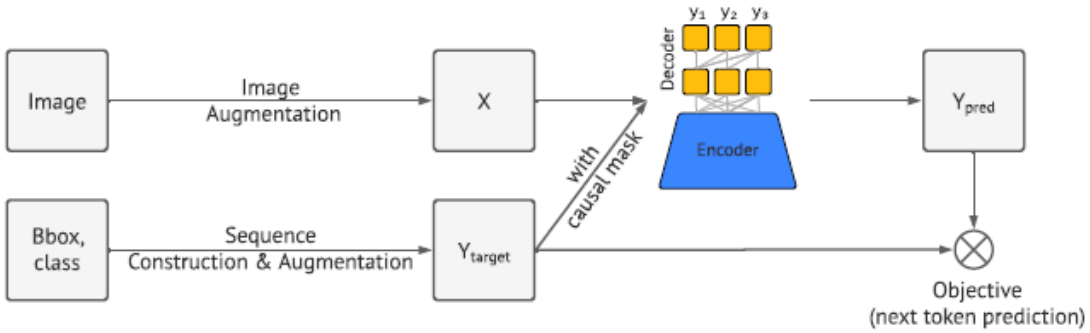
In essence, we cast object detection as a language modeling task conditioned on pixel inputs, for which the model architecture and loss function are generic and relatively simple, without being engineered specifically for the detection task.

For detection task with Pix2Seq:

- quantization and serialization scheme that converts bounding boxes and class labels into sequences of discrete tokens
- leverage an encoder-decoder architecture for perceiving pixel inputs and generating the target sequence

The objective function is simply the maximum likelihood of tokens conditioned on pixel inputs and the preceding tokens.

**Pix2Seq framework**




- Image Augmentation: As is common in training computer vision models, we use image augmentations to enrich a fixed set of training examples (e.g., with random scaling and crops).
- Sequence construction & augmentation: As object annotations for an image are usually represented as a set of bounding boxes and class labels, we convert them into a sequence of discrete tokens.
- Architecture: We use an encoder-decoder model, where the encoder perceives pixel inputs, and the decoder generates the target sequence (one token at a time).
- Objective/loss function: The model is trained to maximize the log likelihood of tokens conditioned on the image and the preceding tokens (with a softmax cross-entropy loss).

**ARCHITECTURE, OBJECTIVE AND INFERENCE**

Architecture:

Use an encoder-decoder architecture. The encoder can be a general image encoder that perceives pixels and encodes them into hidden representations, such as a ConvNet, Transformer, or their combination. For generation, a Transformer decoder will be used.

Objective:

Pix2Seq is trained to predict tokens, given an image and preceding tokens, with a maximum likelihood loss.

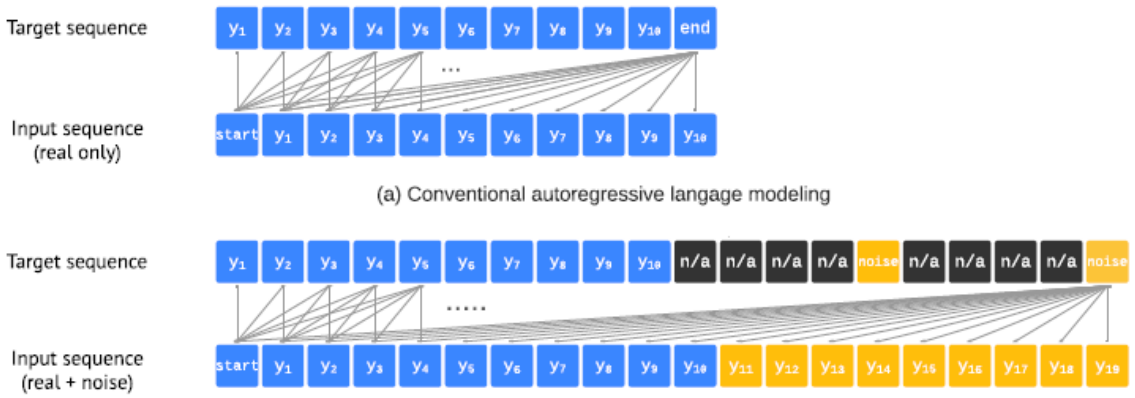
Inference:

At inference time, we sample tokens from model likelihood. This can be done by either taking the token with the largest likelihood (arg max sampling), or using other stochastic sampling techniques. We find that using nucleus sampling leads to higher recall than arg max sampling. The sequence ends when the EOS token is generated. Once the sequence is generated, it is straight-forward to extract and de-quantize the object descriptions.

## SEQUENCE AUGMENTATION TO INTEGRATE TASK PRIORS

The EOS token allows the model to decide when to terminate generation, but in practice, the model tends to finish without predicting all objects. This is likely due to 1) annotation noise, and 2) uncertainty in recognizing or localizing some objects.

To mitigate the problem we simply introduce a sequence augmentation technique, thereby incorporating prior knowledge about the task. The target sequence ~y in conventional autoregressive language modeling is the same as the input sequence y. And all tokens in a sequence are real. With sequence augmentation, we instead augment input sequences during training to include both real and synthetic noise tokens. We also modify target sequences so that the model can learn to identify the noise tokens rather than mimic them. This improves the robustness of the model against noisy and duplicated predictions.



Github links:

Tensorflow: https://github.com/google-research/pix2seq

Pytorch: https://github.com/gaopengcuhk/Stable-Pix2Seq

Paperswithcode link:

https://paperswithcode.com/paper/pix2seq-a-language-modeling-framework-for