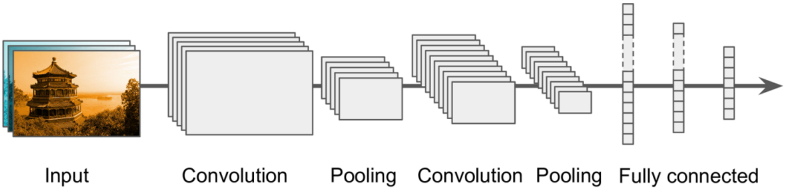
Spatial Pyramid Pooling for Vehicle Detection

Raktim, Rishika, Shreyansh

April 22, 2023

Convolutional Neural Network

- Has multiple iterations of convolution for feature extraction
- These features are then passed on to the next layer, where they are combined and processed
- Subsequent layers may include pooling layers, to reduce the size of feature maps
- After final pooling, conventional fully connected network
- Trained using large labeled images for object detection
- Weights of the neurons are learned to minimize the difference between the predicted output and the true label.

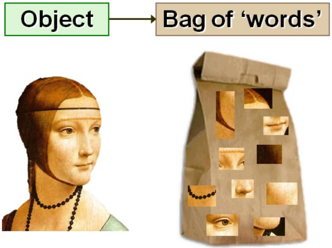


Limitation of CNN for Vehicle Detection

- The convolutional layers of a CNN functions by moving a sliding window over the input image and generates feature maps that capture the spatial layout.
- The following fully connected layers require fixed input image size restricting both aspect ratio and scale of inputs.
- When dealing with images of varying sizes, existing methods adjust the input image to a fixed size through either cropping or warping.
- Can result in geometric distortions or content loss, impacting accuracy.
- Also, using a fixed input size may not be appropriate when objects scales vary.

Bag-of-Words (BoW) Model for Vehicle Detection

- Image is treated as a document and features of image treated as words.
- Features are extracted, codebook is created by clustering extracted features from many images.
- Each image is represented as a bag of visual words by considering the frequency of each visual word in the image.
- A classifier is then trained to distinguish between object classes.



Limitation of BoW for Vehicle Detection

- Sensitive to object scale, orientation.
- Uses a fixed vocabulary, which limits its ability to capture wide range of variations in object appearance.
- Creates a high-dimensional vector to represent an image, making it computationally expensive.
- Doesn't preserve spatial relationships between words, leading to loss of information about object's location and shape.

Spatial Pyramid Pooling (SPP)

- SPPNet is a pooling layer that removes the fixed-size constraint of the network.
- SPPNet can process images of different sizes and aspect ratios and still produce a fixed-length feature vector.
- SPPNet uses multi-level spatial bins, while the sliding window pooling uses only a single window size.
- It pools the features extracted from each sub-region into a fixed-length vector.
- This allows the network to capture information about objects at different scales and positions within the image.

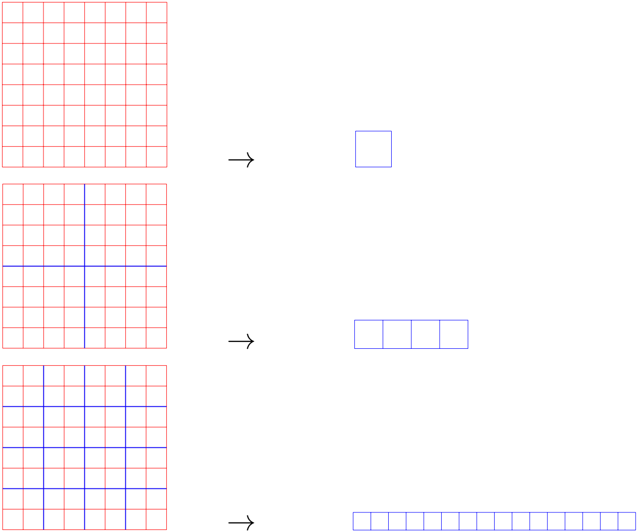
Spatial Pyramid Pooling Layer

- Convolutional layers accept images with varying sizes and aspect ratios, and produce outputs of varying sizes.
- The fixed-length vectors required by the later fully connected layers can be generated using SPP that preserves the spatial information by pooling in local spatial bins.
- Size of spatial bins are proportional to image size but number of bins is fixed irrespective of the image size.
- Replace last pooling layer (after the last convolutional layer) with SPP layer.
- Output of SPP are kM-dimensional vectors, k being the number of filters in last convolutional layer, and M being the number of bins.

How Spatial Pyramid Pooling works

- **Input to an SPP Layer:** Feature map which is the output of the previous convolutional layer in CNN.
- **Dividing the feature map:** The input feature map into a set of rectangular sub-regions at different scales.
- **Pooling features within each region:** Apply max-pooling to each sub-region to obtain a fixed-length feature vector.
- **Concatenating the pooled vectors:** Concatenate the feature vectors from all sub-regions to form the final output.
- **Outputting the feature vector:** The final output of the SPP layer is the concatenated feature vector which is then fed to a fully connected layer for further processing.

How Spatial Pyramid Pooling works



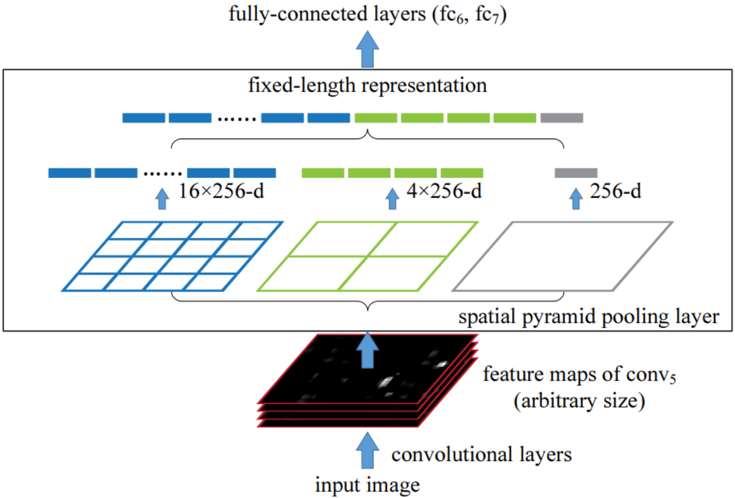
1×1

4×1

16×1

Hence, the final output would be a concatenated vector of dimension 21×1

Spatial Pyramid Pooling Network Architecture



Single-size Training

- Bin sizes required for SPP can be computed for an image of given size.
- For a feature map of size $a \times a$ (e.g., 13×13) after $conv_5$ and a pyramid level of $n \times n$ bins, window size, $win = \lceil a/n \rceil$ and stride, $str = \lfloor a/n \rfloor$.
- We implement l such layers for an $l-$level pyramid, outputs of which are concatenated by next fully connected layer.

[pool3×3]	$[pool2 \times 2]$	$[pool1 \times 1]$
type=pool	type=pool	type=pool
pool=max	pool=max	pool=max
inputs=conv5	inputs=conv5	inputs=conv5
sizeX=5	sizeX=7	sizeX=13
stride=4	stride=6	sizeX=13

[fc6] type = fc outputs = 4096

Multi-size Training

- The primary objective of multi-size training is to simulate the varying input sizes while making use of the existing optimized fixed-size implementations.
- Considering 3 sizes: 180×180, 224×224, and 288×288 resize 224×224 image to 180×180, so that they differ only in resolution.
- We now implement another fixed size input network, here 180×180.
- Feature map size, $a \times a$, $win = \lceil a/n \rceil$ and stride, $str = \lfloor a/n \rfloor$.
- Output of SPP of this 180×180 network has same fixed length as 224×224, i.e, both networks share same parameters.
- One epoch can be trained on one network and then to other one, keeping all weights.