Assignment 4

Notes from sources.

**Pascal visual object classes**

Problem with multi-class datasets:

Problem (i):

Wont work with “which of these m classes does this image contain?”, because the image can contain several instances of the same class and different classes.

Problem (ii):

Classification error. In one picture, you may encounter several non-class examples for every time you detect an actual important class. Using a simple classification error metric is not sufficient.

Problem (iii):

Needing a classification and detection score that is independent of the algorithm used to do classification or detection. Therefore, use some confidence level per image to say how confident you are that an image contains a class (classification), and a bounding box with a confidence level around each object you think are of the correct class per image (detection)

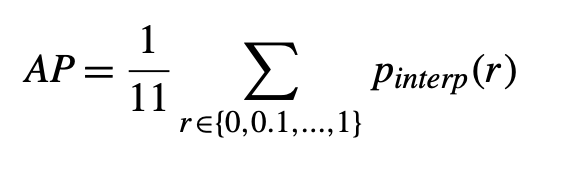
Average Precision (AP) was used for both classification and detection.

For a given task and class we have precision/recall curve.

Recall is defined as the proportion of all positive examples ranked above a given rank.

Precision is the proportion of all examples above that rank which are from the positive class.

AP summarizes the shape of the precision/recall curve, and is defined as the mean precision at a set of eleven equally spaced recall levels [0, 0.1, 0.2, … 1]



Et bilde som inneholder tekst

Automatisk generert beskrivelse

**Bounding Box Evaluation**

In detection tasks, bounding boxes along with confidence levels is used. To measure if a bounding box is correct or not, the object has a pre-defined “correct” bounding box. Then, you compare the overlap of the predicted bounding-box and the pre-defined one.

If the overlap is above 0.5, its interpreted as correct.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

Several detections of the same object counts as one correct detection and several false ones.

**Blog post about mAP**

Calculates average precision value for recall value over 0 to 1.

Precision: measures how accurate your predictions is. The percentage of your predictions that are correct.

Recall: measures how good your model are at finding all the positives.

Et bilde som inneholder tekst, brev

Automatisk generert beskrivelse

**Intersection over union: IoU**

The overlap between two boundary-boxes. We use this metric to determine how much of our predicted box overlaps with the “ground-truth”, that is what is defined as the correct answer.

**AP, average precision:**

Defined as the area under the precision-recall curve.

Given a table showing a models precision and recall in predicting some class, we can draw the precision / recall curve. Precision and recall are always between 0 and 1 => AP also.

Whilst precision can go up and down in value, recall monotonically increase or stay steady. Therefore we get a zig-zag pattern on our curve. So before we calculate the AP we smooth out the curve by replacing the precision value with the maximum one to the right of our current recall value.

Non-smooth curve

Et bilde som inneholder stridsvogn

Automatisk generert beskrivelse

Et bilde som inneholder stridsvogn

Automatisk generert beskrivelse

Smoothing process, the green curve is the smooth curve we get.

This makes the AP value less sensitive to small variations that affect the precision.

**Interpolated AP**

We divide the recall value into 11 points from 0 to 1. Then we calculate the average of maximum precision value for these 11 values.

Et bilde som inneholder stridsvogn

Automatisk generert beskrivelse

Et bilde som inneholder logo

Automatisk generert beskrivelse

“the intention in interpolating the precision/recall curve in this way is to reduce the impact of the wiggles in the precision/recall curve, caused by small variations in the ranking of examples.”

However, this method suffers in accuracy and its lost its capability in measuring difference for methods with low AP. Since low AP gets filtered away by looking for the max value.

**AP AUC, average precision area under curve**

Et bilde som inneholder stridsvogn

Automatisk generert beskrivelseAfter smoothing the curve we now look at the area under the curve. Instead of sampling 11 points we sample p(r) whenever it drops and computes AP as the sum of the areas under the curves.

Et bilde som inneholder tekst, brev

Automatisk generert beskrivelseThis is the AUC.

**Mean average precision mAP**

The average of AP. We compute the AP for each class and average them.

**Blog post about SSD**

**SSD – Single Shot Detection**

Consists of two parts:

* Extract feature maps
* Apply convolution filters to detect objects

VGG-16; a convolutional network architecture that is 16 layers deep

Conv4\_3; The identification of convolutional layer. Name

A Conv4\_3 layer makes 4 object predictions. Each prediction composes of a boundary box and 21 scores for reach class, and we pick the highest score as the class for the bounded object. So a Conv4\_3 layer makes 38x38x4 predictions, given that the image is 38x38.

No object in cell has its own reserved “0” class.

SSD computes the location and class scores using small convolutional filters. After extracting the feature maps SSD applies 3x3 convolution filters for each cell to make predictions.

Each filter outputs 25 channels: 21 scores for each class, plus one boundary box. Et bilde som inneholder stridsvogn

Automatisk generert beskrivelse

For example, in conv4\_3 we apply four 3x3 filters to map 512 input channels to 25.

SSD use multi scale feature maps to detect object independently. As the CNN reduce the spatial dimension gradually, the resolution of the feature maps also decrease. So SSD use lower resolution layers to detect larger scale objects.

This makes the network able to detect objects of different scales in the same image.

After the VGG16 network, SSD have another 6 convolution layers. Five of them are added for object detection. Three of those make 6 predictions.

**Default boundary box**

The default boundary box, and how to define them represents a challenge. Using only one, example standing box, it could be good for detecting a human, however a car might need a wider box. Also, placing the “correct” bounding box is more difficult if we only have one size and aspect ratio to use.

Training the model with one default box, early training is very unstable.

Therefore we opt to use more shapes of boundary boxes and sizes. Boxes that fit one class better gets a higher IoU score and is kept.

We can group the ground truth boundary boxes into clusters, where each cluster have one default boundary (might be standing aspect ratio that fits most humans) at the center. Instead of making random guesses as to how the boundary box should look like, we start with the defaults and work from there.

SSD often have a lower number of pre-defined default boxes, 4-6, and make one prediction per default box.   
the defaults a predefined to fit most object you might find in the dataset. So, example standing with some aspect ratio for humans, and landscape with some specific aspect ratio for cars and such.

Then instead of looking at each predictions location globally, we look at them relative to their respective default boundary box.

Each specific feature map layer shares the same set of default boxes, but different layers use different sets of default boxes to customize detection of the different resolutions through the CNN.

**Choosing default boundary boxes**

They are chosen manually for SSD. Define a scale value for each feature map layer. Starting from the left, Conv4\_3 detects objects at the smallest scale at 0.1 or 0.2. then increases linearly to the rightmost layer at a scale of 0.9.

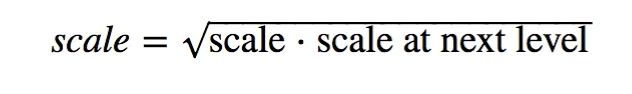
The scale combined with the target aspect ratio we get the width and height of the default bounding box.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

For a layer making 6 predictions, SSD starts with 5 target aspect ratios; 1, 2, 3, ½, 1/3.

Then adds an extra default box with scale:



And aspect ratio = 1

**Matching strategy**

SSD define a prediction as positive or negative match. If the corresponding **Default** **Boundary** **Box** has an IoU greater than 0.5 with the ground truth, the match is positive. Otherwise, negative.

Once we identify the positive matches with **default** boundary boxes, we use the corresponding **predicted** boundary boxes to calculate the cost. This makes the network approach some universally defined best matching default boxes instead of some interpreted box placement.

**Multi-scale feature maps & default boundary boxes**

Different size objects are detected at different resolutions. Smaller objects are detected at higher resolutions. So higher resolution feature maps are responsible for detecting small objects. SSD are therefore worse in general at detecting smaller objects, this is compensated by using higher resolution images.

**Loss function**

The **localization loss** is the mismatch between the ground truth box and the predicted boundary box. SSD only penalizes predictions from positive matches. Negative matches are ignored.

The **Confidence loss** is the loss of making a class prediction. For every prediction positive match, we penalize the loss according to the confidence score of the corresponding class. For negative match predictions we penalize the loss according to the confidence score of the class “0”.

**Hard negative mining**

For every image we make far more negative prediction than positive ones. This is due to the background is in general not part of the class we want to guess. So there will be many more negative than positive matches. This approach trains the network to detect backgrounds. We still need negative predictions though. So the solution is picking the negatives with the top loss and makes sure the ratio between the picked negatives and positives is at most 3:1.

Task 1

1. The intersection over Union is a method of checking how much of a predicted bounding-box overlap with a pre-defined correct one. If this number is higher than some threshold, its considered as a correct guess. Its argued that a relatively low number should be used to account for ambiguous bounding boxes. For example, one could argue over what the correct bounding box for a person, spreading his or hers arms and legs should be.

For two bounding boxes we look at the amount of area that’s in both boxes (higher is better), and divide by the area that the two boxes combined cover (lower is better). Therefore we get the Intersection/Union method.

Et bilde som inneholder tekst

Automatisk generert beskrivelse

From the drawing we can see two things:

1. If the intersection is low, the union becomes bigger, and we get a small number divided by big number, in the end resulting in an even smaller number. Bad.
2. If the intersection is close to the whole bounding-box size, the union approaches the area of the bounding box also. Therefore the division approach 1 = 100% overlap.
3. Precision: TruePositives / (TruePositives + FalsePositives)

Recall: TruePositives / (TruePositives + FalseNegatives)

Given a positive and a negative class:

True Positive is when we say something is of a positive class, and it is.

False Positive is when we say something is of a positive class, but it is negative.

1. Mean average precision:

Et bilde som inneholder diagram

Automatisk generert beskrivelse

Task 3

1. Non-maximum suppression
2. True, deeper layers have a lower resolution, and are therefore better at detecting more general object spanning a greater portion of the image.
3. The reason SSD uses different bounding box aspect ratios at the same spatial location is to detect objects with different aspect ratios. Different objects in an image can have different aspect ratios, and using only one aspect ratio for each location in the feature map may result in poor performance on objects with aspect ratios that do not match the chosen ratio. By using multiple aspect ratios for each location, the network can detect objects with different shapes and aspect ratios. In addition to aspect ratio, the network also uses different scales for the bounding boxes to handle objects of different sizes. This combination of different aspect ratios and scales allows the network to generate a large number of bounding boxes with varying sizes and shapes, which improves the accuracy of object detection.
4. The main difference lies in the way they generate bounding box proposals. YOLO divides the input image into a grid and predicts the class probabilities and bounding box coordinates for each grid cell. Each grid cell only predicts one bounding box, even if multiple objects are present within it. In contrast, SSD generates multiple anchor boxes with different scales and aspect ratios at each spatial location in the feature map and uses these to predict the class probabilities and offsets for the corresponding objects. This means that SSD generates more bounding box proposals than YOLO, which allows it to detect objects of different sizes more accurately.
5. 38x38x6 = 8664 anchor boxes

38 x 38 x 6 = 8664

19 x 19 x 6 = 2166

10 x 10 x 6 = 600

5 x 5 x 6 = 150

3 x 3 x 6 = 54

1 x 1 x 6 = 6

8664 + 2166 + 600 + 150 + 54 + 6 = 11640

Total = 11640