Exam-solution - Exam-solution

Computer Vision III: Detection, Segmentation and Tracking (Technische Universität München)

Chair of Computational Modeling and Simulation
Department of Civil, Geo and Environmental Engineering
Technical University of Munich

Esolution
Place student sticker here

Note:

- During the attendance check a sticker containing a unique code will be put on this exam.
- This code contains a unique number that associates this exam with your registration number.
- This number is printed both next to the code and to the signature field in the attendance check list.

Computer Vision III: Detection, Segmentation and Tracking

Exam:	IN2375 / Endterm	**Date:**	Wednesday 21st July, 2021
Examiner:	lealtaixe	**Time:**	11:30 – 13:00

Working instructions

- This exam consists of **12 pages** with a total of **3 problems**.
  Please make sure now that you received a complete copy of the exam.
- The total amount of achievable credits in this exam is 60 credits.
- Detaching pages from the exam is prohibited.
- Allowed resources:
  - –
- Subproblems marked by * can be solved without results of previous subproblems.
- **Answers are only accepted if the solution approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.itemize

Problem 1 Multiple Choice (12 credits)

Mark your answer clearly by a cross in the corresponding box. Multiple correct answers per question possible. For every question, you will either get full credit (if you mark all the correct answers) or no credit (if you do not mark at least one of the correct answers).

Mark correct answers with a cross ☒
To undo a cross, completely fill out the answer option ■
To re-mark an option, use a human-readable marking ✕■

a) Check all that apply for segmentation:

- [ ] Semantic segmentation is equivalent to instance segmentation.
- [ ] Panoptic segmentation can be considered as a combination of semantic and instance segmentation.
- [ ] Semantic segmentation must use dilated convolutions.
- [ ] Mask R-CNN performs semantic segmentation.

b) Mean Average Precision (mAP) is used to evaluate object detectors. Which of the following statements is true (mark all that apply):

- [ ] Average Precision (AP) does not change with the number of low-score False Positives (FP) when all True Positives are covered by high-score predictions.
- [ ] Average Precision is also called the Jaccard coefficient.
- [ ] First the Average Precision is computed for each class, then the mean of AP is taken over all classes.
- [ ] First the Average Precision is computed for each bounding box, then the mean of AP is taken over all boxes.

c) Check all that apply for object detectors:

- [ ] YOLO is an one-stage detector while SSD is a two-stage detector.
- [ ] DETR does not use positional encoding.
- [ ] R-CNN does one forward pass through the backbone CNN for every proposal.
- [ ] Fast R-CNN uses anchors.

d) Check all that apply for metric learning:

- [ ] Metric learning is the task of classifying images.
- [ ] Contrastive loss uses all the relations in the mini-batch.
- [ ] Number of classes in the training set must be the same as in the testing set.
- [ ] Triplet loss modifies contrastive loss, so it uses more relations.

e) Which of the following statements is/are true about Message Passing Networks (check all that apply):

- [ ] They can be implemented as graph neural networks.
- [ ] They are invariant to node permutations.
- [ ] They cannot be trained end-to-end.
- [ ] They use node embeddings and might use edge embeddings.

f) Which of the following statements is/are true about Transformers (check all that apply):

- [x] Transformers use an encoder-decoder architecture.
- [ ] The number of layers in the encoder must be the same as the number of attention heads in the attention layers.
- [x] In Transformers, the output of multi-head attention is fed into an MLP.
- [ ] Positional encoding is learned in Transformers.

Problem 2 Short questions (28 credits)

0 1 2 3

a) What is the main problem of R-CNN networks (Girshick et al., CVPR 2014) (1p)? In an image containing N proposals, what is the needed number of forward passes (1p)? Briefly give a solution to this problem (1p)?

The main problem of R-CNN networks is that they are slow. The region proposals are not learned end-to-end but defined by a handcrafted algorithm (e.g., selection search). (1p) Then each region proposal is passed to a CNN, thus for N region proposals, we need to do N forward passes over the CNN. (1p)
A possible solution is to first pass the image into the CNN, and then find the regions of interest in the feature space, thus making the model more efficient, as is done in Fast R-CNN. (if the student gives the Faster R-CNN solution instead, that also gives full credit). (1p)

0 1 2

b) Write the equation of multi-head attention (1p). Clearly describe every term in the equation (1p).

First the matrix of queries is multiplied with the matrix of keys and is passed over a softmax function (d_k is just a scaling factor). In this way, you get a probability distribution that tells each feature where to attend more (the higher the softmax value, the more it attends). Then, this product is multiplied with the matrix of values, and this can be interpreted as soft-indexing. (1p) If equation is totally wrong, just describing the terms does not give any point.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

(1p) for correctly writing the equation.

0 1 2 3

c) What is Masked Multi-Head attention and how does it differs from Multi-Head attention (1p)? In what part of the Transformer you use Masked Multi-Head attention (1p) and in what part you use Multi-Head attention (1p)?

Transformers are autoregressive models, so their job is to learn $P(x_i|x_{j<i};\theta)$. For efficiency concerns, we feed the entire sequence $_{1\ldots n}$ into the model, and then just implement things correctly so that the model does not look at any x_j for $j \geq i$ when computing $P(x_i|x_{j<i};\theta)$. This is called masked multi-head attention. (1p)
You use masked multi-head attention at the beginning of a transformer layer in the decoder (1p) (no points if only mentioned the decoder).
You use multi-head attention in all Transformer layers of the encoder, and the second part of the decoder (1p) (no points if they mention only the encoder).

d) In the lectures, we presented a type of detection network that does not need to use non-maximum supression. What type or architecture the network uses (1p)? Name another task (in addition to object detection) that the network can be used for (1p).

0 1 2

DETR network uses an encoder-decoder architecture that contains a CNN followed by a Transformer network. (1p, it is okay to not mention the name of the network)
In addition to object detection, the network can be used for Panoptic Segmentation (1p, must specify panoptic part to get full credit, if it is just segmentation that does not give points).

You are given an undirected graph $G = (V, E)$ with 5 nodes, $V := 1, 2, 3, 4, 5$ and edges $E = [(1, 3), (2, 3), (3, 4), (3, 5)]$. Every node i corresponds to an object detection observed at a time step i. Neural message passing is being performed on G. Assume that a round of *node-to-edge* updates has been performed, and you have access to the intermediate *messages*: $m(1, 3) = (1, 0)'$
$m(2, 3) = (0, 1)'$
$m(3, 4) = (1, 1)'$
$m(3, 5) = (0, 0)'$

e) For the *edge-to-node* updates, we are using the *mean* operator. Perform a regular *edge-to-node* update on node 3 (1p).

0 1

Solution: (1 / 4) * (1 + 0 + 1 + 0, 0 + 1 + 1 + 0)' = (0.5, 0.5) ' (1p)

f) You are given an additional 4x2 weight matrix $N_v = ((1, 1, 1, 1), (1, 1, 1, 1))$. Perform a time-aware *edge-to-node* update on node 3 (2p).

0 1 2

First, we compute the mean separately for messages from nodes in past frames (1 and 2), and nodes in future frames (4, 5):
$(1/2) * (1 + 0, 0 + 1)' = (0.5, 0.5)'$ $(1/2) * (1 + 0, 1 + 0)' = (0.5, 0.5)'$ (1p if both computations correct)
Now, we concatenate these two vectors and multiply the result with matrix N_v: $((1, 1, 1, 1), (1, 1, 1, 1)) * (0.5, 0.5, 0.5, 0.5) = (2, 2)'$ (1p)

0 1 2 3 4

g) Write the equation for the refinement in The Group Loss paper (1p), make sure to explain all the variables. What is the purpose of the numerator (1p)? What is the purpose of the denominator and why is it needed (1p)? What is the main conceptual difference to softmax function typically used in neural networks (1p)?

$$x_{i\lambda}(t+1) = \frac{x_{i\lambda}(t)\pi_{i\lambda}(t)}{\sum_{\mu=1}^{m} x_{i\mu}(t)\pi_{i\mu}(t)}$$

$x_{i\lambda}(t)$ is the matrix of priors (probability matrix) at time step t. $\pi_{i\lambda}(t)$ represents the support that the current mini-batch gives to the hypothesis that the i-th image in B belongs to class λ.
It refines the matrix of priors considering the similarities between all the mini-batch images, as encoded in the similarity matrix, as well as their labeling preferences. (1p)
The numerator propagates the information between the samples in the minibatch (1p).
The denominator is a normalization term that ensures that the labeling remains on the standard simplex (probability space) (1p).
The main conceptual difference to softmax function, is that while softmax function considers only the local information (the features of a sample), the refinement procedure in the Group Loss combines the local information (priors) with the global information (the similarity between all pairs on the minibatch) (1p).

0 1

h) A weakness of Group Loss is that samples communicate only by sharing their labeling preferences. Can you give an extension of it in order to allow samples sharing their feature information too? Briefly justify your answer. (1p)

Samples can communicate via Message Passing Networks to share their feature information. This can be done by using Transformers.

0 1 2

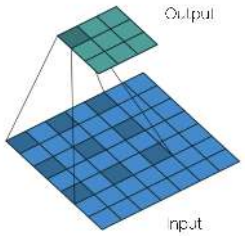
i) Explain the 3 training steps of OSVOS (S. Caelles et al. “One-shot video object segmentation”. CVPR 2017), detailing the purpose of each step (2p).

1. Pre-training: the network is pre-trained on ImageNet for the task of object classification. (0.5p)
2. Parent network training: the network is trained for the general task of video object segmentation with all the training set sequences. (0.5p)
3. Fine-tuning: the network is fed with the first frame ground truth mask of the object. It is then fine-tuned on this mask (plus data agumentation of the mask), in order to learn its appearance and to know *which* object to segment in the next frames. (1p).

j) You are doing semantic segmentaion and want to increase the size of the receptive field without increasing the number of weights in your network. How can you do so (explain the main concept behind your solution) (1p)? Illustrate your solution by either writing the equation or drawing an example (1p).

0 1 2

This can be done by using atrous (dilated) convolutions. On this case, the convolutions weights are spread out (dilated), so that we can increase the receptive field of the convolution without increasing the number of weights or parameters. (1p)



(1p given for drawing the figure correctly and/or writing the equation and/or explaining in detail the concept of dilated convolutions).

k) Explain the difference between semantic and instance segmentation (1p). Briefly describe a network that does both semantic and instance segmentation at the same time (1p).

0 1 2

On semantic segmentaion, you are interested in classifying each pixel to the class it belongs. It makes no difference between pixels coming from different instances of the same class. For example, in an image with two cars (car A and car B), semantic segmentaion will just classify all the pixels coming from cars as 'car' pixels. It is also able to segment non-countable objects like sky or river.
Instance segmention, on the other hand differentiates between pixels belonging to the same class but belong to different objects (for example, pixels coming from car A, or pixels coming from car B), and it cannot segment the uncountable objects. (1p)
These concepts got unified in panoptic segmentation, that is able to do both semantic and instance segmentation as the same time. Such a network is UPSNet, that does semantic segmentation for the uncoutable objects (stuff) and instance segmentation for the countable objects (thing). (1p, no point given if the student only mentions the name of the network without describing it.)

l) UPSNet (Xiong et al., CVPR 2019) uses a special type of convolutions in their semantic segmentation head. Write and explain the equation behind this concept (1p). Explain why they are important on this type of network (1p).

0 1 2

Regular convolution

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n)$$

Deformable convolution

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \Delta\mathbf{p}_n)$$

where $\Delta\mathbf{p}_n$ is generated by a sibling branch of regular convolution

Deformable convolutions generalize the concept of dilated convolutions by also learning the offset, which is generated by a sibling branch of the regular convolutions. (1p)
The main advantage of them is that they can pick values at different locations for convolutions conditioned on the input image of the featured space. Thus, it is not limited to rectangular shapes that standard convolutions use, making it more precise especially near the boundaries of objects with non-regular shapes. (1p)

0 1 2

m) Explain the major problem of training a deterministic (one-to-one mapping) Deep Learning Model (e.g. S-LSTM) with an $L2$ loss for pedestrian trajectory prediction (1p). How could you extend the model to solve this problem (1p)?

Problem of trajectory prediction is multimodal/ stochastic - Optimisation of L2 loss results in unrealistic average path/ linear trajectory (1p)
Using generative (GAN, VAE) or stochastic model (e.g. Gaussian mixture model) (1p).

Problem 3 **Long question (20 credits)**

You are building part of the vision pipeline that should be built into a car to provide it with autonomous driving capabilities, and your task is to perform multi-object tracking of *cars* and *pedestrians*, for which you have enough training data. For now you are building a prototype, so you are not concerned with computational time.

a) You decide to follow a tracking-by-detection paradigm. Explain the steps of such a methodology for tracking (1p).

0 1

In tracking-by-detection, the multi-object tracking problem is divided into two steps: (i) first an object detector is applied to each frame of the image; (ii) a data association step links detections belonging to the same trajectory across frames.

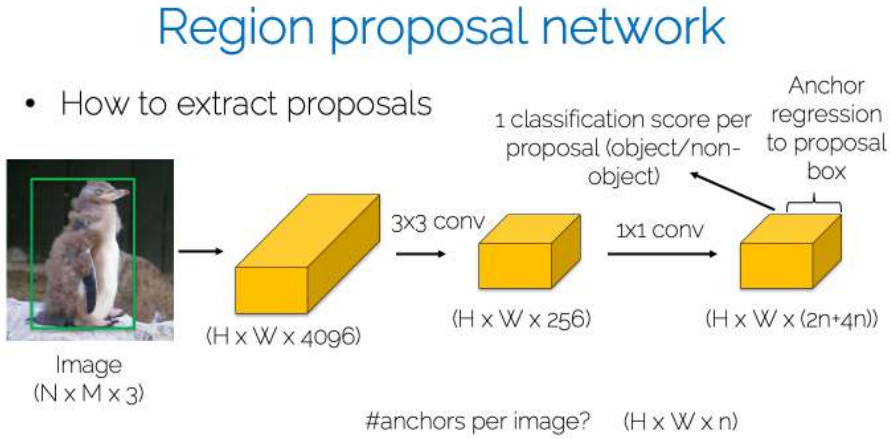
b) What is the key difference between the tracking-by-detection and the tracking-before-detection paradigm (1p)?

0 1

Tracking before detection aims to initialise tracks based on category-agnostic bounding box representation of the input signals. This is based on generic objectness cues, such as spatial proximity, motion consistency and appearance similarity. At this stage, we do not know to which semantic classes our tracks belong. In contrast, in case of tracking-by-detection, we only track objects that we have already recognised (classified). (Stating we first track the objects with a motion model and then "detect" them does not give points, this is pretty much just expanding the wording "tracking-before-detection". The notion that the class is not know, therefore, we need to run the classification after tracking needs to be stated to get the points).

c) To perform object detection, you implement a version of *Faster RCNN* (Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015). Draw the architecture of a Region Proposal Network (RPN) as seen in *Faster RCNN*. Make sure to specify the sizes of all the feature maps and your convolution kernels. Use H and W as the size of the feature map, C as the number of channels (use C_1, C_2 or as many as you need) and n as the number of anchors per location on the feature map (2p). How are proposals extracted from the RPN, i.e., what do the numbers in the last feature map represent (2p)?

0 1 2 3 4



n = number of anchors
2n = used to classify the object as object/non-object (1n also accepted)
4n = bounding box regression for each anchor

Your code is now trained to detect cars and pedestrians, so you move to the temporal domain. You now want to track these objects, so you decide to follow *Tracktor* (P. Bergmann et al. "Tracking without bells and whistles". ICCV 2019).

0 1 2

d) Explain what elements of your previous object detector can *Tracktor* re-use and how, in order to obtain tracks of cars and pedestrians from a video (2p).

You can re-use the bounding box regressor. (1p)

Instead of using the RPN at each frame, you input the previously detected bounding boxes of frame $t-1$ as proposals for detection at frame t. Using the regressor, you now know where those boxes moved from frame $t-1$ to frame t, effectively solving the tracking problem for two frames. You repeat this procedure for every new frame. You also ran the plain detector in parallel in case pedestrians appear in the scene. (1p)

0 1 2 3

e) Your implemented *Tracktor* idea is working well in tracking isolated objects, but as soon as you have a crowded scene with many pedestrians, you observe a lot of *identity switches*. You decide to train a re-identification network for similarity learning. What architecture (with a ResNet-50 backbone) (1p) and loss (1p) do you use? Write down the formula for that loss (1p).

Architecture: siamese network. (1p)
Loss: Triplet loss (contrastive loss is also accepted) (1p)

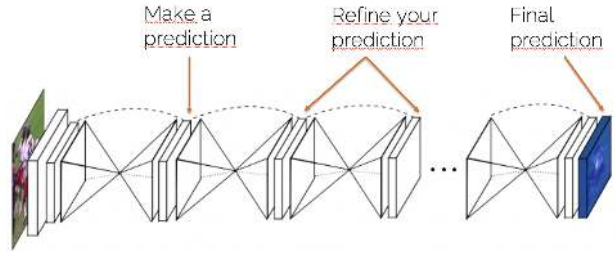
Formula: $\mathcal{L}(A, P, N) = \max(0, |||f(A) - f(P)||||^2 - |||f(A) - f(N)||||^2 + m)$, where A indicates the anchor image, P the positive image, N the negative image and m the margin. (1p)

NB: If the students use some other metric learning loss, they also get full-credit assuming that the formula matches the loss they mention.

0 1 2 3

f) Your tracker is now quite stable, but you see the re-identification could be improved if we had body joint locations. You decide to reimplement the stacked hourglass architecture for human joint prediction. What is the output of such architecture and how does it represent joint locations (1p)? Explain the architecture (draw an example) and why is is called *stacked hourglass* (1p). In which part of the architecture do you compute the loss (1p)?

Representation: body joint locations as heatmaps, same dimension as an image but with high values on the position where each of the joint is likely to be found. (1p)



It is called stacked hourglass because we have several u-net (encoder-decoder which look like an hourglass) concatenated. The series of downsampling and upsampling steps helps us refine the predictions. (1p)
The loss is computed at all intermediate outputs in order to get a stronger training signal (1p).

g) What is the main evaluation metric for multiple object tracking (0.5p)? State its formula and what each of the terms means (1.5p)?

0 1 2

MOTA = multiple object tracking accuracy. (0.5p)

Multi-object tracking accuracy → $\text{MOTA} = 1 - \frac{\sum_t (\text{FN}_t + \text{FP}_t + \text{IDSW}_t)}{\sum_t \text{GT}_t}$, ← Ground truth

In the numerator you count the number of errors (false positives, false nagatives and identity switches) while in the denominator you have the number of ground truths. The lower the number of errors, the lower the expression will be. Finally, you compute MOTA by subtracting the expression from 1. Consequently, the lower the number of errors, the higher is the overall MOTA score. (1.5p)

h) You now want to scale your detection to all possible object classes in the world. Is that feasible or not? Explain your argument (1p).

0 1

It is not feasible because of the long tail problem – several objects are observed far too infrequently to collect a sufficient amount of training data for them.

i) Even if you have access to only video frames, you want to develop a 3D multi-object tracking method. Note, you do not have access to LiDAR data. Propose an approach with which we can leverage the components discussed in the lecture without modifications. (1.5p) Argue why your choices are the best. (1.5p)

0 1 2 3

The approach:
1. Predict the depth of the scene (e.g., with a monocular depth neural network). (0.5p)
2. Convert depth maps to point clouds, treat them as a LiDAR signal. (0.5p)
3. Use existing 3D object detectors on this representation (re-training will be required though). PointRCNN would be ok, but possibly we would like to use detector that also uses image data, not only relying on point clouds. (0.5p)

Why these choices are the best? For tracking, I would not recommend to go with AB3DMOT because it is relying on precise 3D localization too much. GNN3DMOT would be a better approach, as it leverages both images and ?3D signal?, where 3D signal in this case will be point cloud representation of the estimated depth map. (1.5p)

Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.