INFOMCV 2021 practice exam

Cursus: BETA-INFOMCV Computer vision (INFOMCV)

Inhoud: Pagina's:

Aantal vragen:	6
Gegenereerd op:	7 apr. 2021

INFOMCV 2021 practice exam

Cursus: Computer vision (INFOMCV)

This is a practice exam. You are allowed to use all offline and online materials and other resources but you are not allowed to communicate with anyone about the contents of the exam during the exam period.

Aantal vragen: 6

In totaal zijn 54 punten voor deze toets te behalen, 27 punten zijn nodig om voor de toets te slagen.

1 You intend to develop an augmented reality app for iPhones. To this end, you need to know exactly
4 pt. the position and orientation of your iPhone (camera) in a room. Your solution is to hang a chessboard on the wall and require that it is always in view for the augmented reality to work. For each iPhone type, you have the camera intrinsics calculated.

Explain what (data, parameters, functions, etc.) is needed to make this app work. Discuss the **offline phase and online phase**.

2 We would like to implement the silhouette-based voxel reconstruction algorithm but don't want to rely on background subtraction. Instead, we have a set of 100 pairs (T_{HOG}, T_{mask}). Each T_{HOG} is a HOG template that is normalized to unit length. Each T_{mask} is a binary mask image with zeros corresponding to the background and ones corresponding to the foreground. The templates have been extracted from a similar setup with only people in the foreground. Each mask is hand-annotated and has the same size: 30x80. The HOG descriptor has been calculated over the same image area as the mask.

Develop a pipeline (sequence of processing steps) that generates a voxel model based on the input of four calibrated cameras. Each frame is a 640x480 RGB image. People in each frame are between 80 and 160 pixels tall. Instead of using background substraction, use the templates. You may assume that there is a function *projectToView(voxel_ID, cam_ID)* that outputs the 2D pixel coordinate where voxel_ID projects to in view cam_ID. So you don't have to construct the look-up table.

8 pt. **a.** Use pseudo-code or text to describe your pipeline, **assuming there are no false positives** when matching the templates. Clearly explain/initialize your data structures. If there are parameters in your approach, mention these explicitly and explain what they do. From your text, I should be able to reproduce your pipeline.

5 pt. **b.** In case we could have **false positives when matching the templates**, how can we adapt this pipeline? Again, explain and initialize your data structures and describe additional parameters.

3 We're forming a start-up company that sells hardware-software integrated computer vision solutions to a growing market of chess players. In the assembled setup, we have a top-down camera system directly above a chess board. The functionality to undistort the view and to detect the boundary of the 8x8 play field is ready. Also, we can find the crossings of the tiles (cells). Based on this functionality, we can cut a part from the image that covers a specific tile.

Our next step is to recognize whether there is a piece on each tile and, if so, which piece (color and type). There is a significant amount of variation in the appearance due to the different sets of chess pieces, but also due to lighting conditions and rotation. We thus resort to a supervised learning-based method using SIFT. In the online phase, we extract SIFT descriptors in the 2D image area for each tile. You can assume there are at least 3 descriptors found for each non-empty tile. Now we turn to the offline phase, to train the classifier.

3 pt. **a.** For this application, **motivate** which of the following data augmentation techniques are useful: horizontal mirroring, rotation (-30...30 degrees), contrast adjustment (-20...20%).

7 pt. **b.** Explain how we can generate a bag-of-words codebook given the training set. We have 250 images of a chessboard with varying numbers of pieces available. You may assume that the label for each tile of each image is provided. The number of codewords in our codebook is 100. Use pseudo-code or text to describe the process. Clearly explain/initialize your data structures. If there are parameters in your approach, mention these explicitly and explain what they do.

4 We focus on underwater reef images such as the one below. We'd like to count the fish. To this end, we have a black-box classifier function *classifyRegion* that takes a bounding box and an image and returns a 1 when there is a fish and 0 when there is no fish.

Our job is to develop an algorithm that generates all bounding boxes that tightly surround single fish instances. To this end, we use the Selective Search algorithm. We start with each pixel as a separate region. Iteratively, we merge neighboring regions. The criterion for merging regions is the Euclidian distance in the hue-saturation space between the means of all pixels in each of the two neighboring regions. We ignore the value channel. At each iteration, we increase the threshold for merging and merge when the distance is below this threshold.

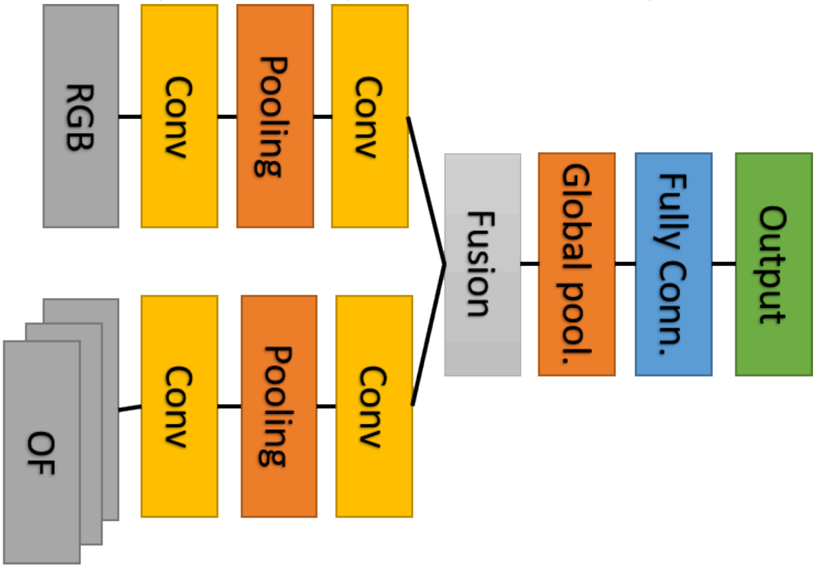
At the end of each iteration, we calculate the bounding boxes to be the minimum enclosing rectangles of each region. We then feed each region to our *classifyRegion* function and store the returned output together with the bounding box. We compare this list of (returned output, bounding box) pairs to a similar list with manually annotated bounding boxes for all observable fish in the image. Based on these two lists and a given IoU threshold, we determine the precision and recall using a predefined IoU threshold of 0.5.

4 pt. **a.** What happens to the precision and the recall when the iteration number increases? Motivate your answer.

4 pt. **b.** We stop half-way the merging process. What happens to the values for precision and recall when we gradually increase the IoU threshold from 0 to 1? Motivate your answer.

5 We will develop a Two-stream CNN to classify videos of a hand gesture (5 classes: thumbs up, wave, point, peace, and fist). The input consists of (1) a 64x64x3 input image and (2) a stack of 8 two-channel (horizontal and vertical) optical flow images of size 40x40. Each input is processed by a branch of convolution, pooling, and a second layer of convolution. Then we perform mid-level fusion by concatenating the two activation volumes after the second pooling layer. Based on this fusion, we perform global pooling, followed by a single fully-connected layer of size 100 and an output layer. See image below for our two-stream CNN architecture.

The number of filters in each convolution layer is 16. Your network should, realistically, be able to learn relevant patterns with expected variations in the input.



5 pt. **a.** Provide for each layer the relevant hyper-parameters (e.g., size, stride, padding, number of filters, type of pooling). Make reasonable choices (e.g., no 51x51 convolutions). No need to motivate your choices.

5 pt. **b.** Provide for each layer the number of parameters and the size of the output activation volume. Also provide your calculation. No need to motivate your answer.

6 You lost all the photos on your phone and didn't make back-ups. Luckily, you posted e-ve-ry-thing on Instagram. You always used the same color filter to make it look more vintage. Now you want to recover your original photos, so without the color filter applied. You don't panic because you took the computer vision course and learned about CycleGANs.

4 pt. **a.** Given that you still have your phone, your Instagram account and your color filter, explain how you could get your original photos back (as well as possible). Explicitly mention how you obtain your training data, and provide a step-by-step explanation of how the CycleGAN is applied for your purpose.

5 pt. **b.** While working on this solution, you have been playing around with a personal cat recognizer you once implemented. When applying it to your Instagram photos, you notice that many mistakes are made. Your cat detector is a CNN based on the Inception V1 architecture, but with 8 output classes for all the cats you ever had. The input to your CNN is a crop of a cat, scaled to a standard size of 224x224. For all your photos, you have manual annotations of the bounding boxes of all the cat instances. Instagram didn't change the resolution so the annotations are still valid.

You plan to use transfer learning to make sure you can also recognize your beloved cats in your Instagram photos. But, instead of freezing the lower layers, you perform weight updates of all parameters. Explain step-by-step how we can generate training data, do the transfer learning and develop the final classifier for the Instagram inputs.

Naam:			Handtekening:
Datum:	- -	Geboortedatum:	- -

Cursus: BETA-INFOMCV Computer vision (INFOMCV) - Vragen: INFOMCV 2021 practice exam

This is a practice exam. You are allowed to use all offline and online materials and other resources but you are not allowed to communicate with anyone about the contents of the exam during the exam period.

1
4 pt.

Antwoord:

2

13 pt. **a.**

Antwoord:

b. Antwoord:

3
10 pt. **a.**

Antwoord:

2

4

6

8

b. Antwoord:

4
8 pt.

a. Antwoord:

b. Antwoord:

5

10 pt. **a.**

Antwoord:

b. Antwoord:

6

9 pt. **a.**

Antwoord:

b. Antwoord:

Correctiemodel

1.
4 pt.

Correction criterion	**Points**
Criterium 1	4 points
Total points:	*4 points*

2.
13 pt.

a.

Correction criterion	**Points**
Criterium 1	8 points
Total points:	*8 points*

b.

Correction criterion	**Points**
Criterium 1	5 points
Total points:	*5 points*

3.
10 pt.

a.

Correction criterion	**Points**
Criterium 1	3 points
Total points:	*3 points*

b.

Correction criterion	**Points**
Criterium 1	7 points
Total points:	*7 points*

4.
8 pt.

a.

Correction criterion	**Points**
Criterium 1	4 points
Total points:	*4 points*

b.

Correction criterion	**Points**
Criterium 1	4 points
Total points:	*4 points*

5.
10 pt.

a.

Correction criterion	**Points**
Criterium 1	5 points
Total points:	*5 points*

b.

Correction criterion	**Points**
Criterium 1	5 points
Total points:	*5 points*

6.
9 pt.

a.

Correction criterion	**Points**
Criterium 1	4 points
Total points:	*4 points*

b.

Correction criterion	**Points**
Criterium 1	5 points
Total points:	*5 points*

Cesuur

Toegepaste raadscore: 0 pt

Behaalde punten	Cijfer
54	10
53	9,8
52	9,7
51	9,5
50	9,3
49	9,2
48	9,0
47	8,8
46	8,7
45	8,5
44	8,3
43	8,2
42	8,0
41	7,8
40	7,7
39	7,5
38	7,3
37	7,2
36	7,0
35	6,8
34	6,7
33	6,5
32	6,3
31	6,2
30	6,0
29	5,8
28	5,7
27	5,5
26	5,3
25	5,2
24	5,0
23	4,8
22	4,7
21	4,5
20	4,3
19	4,2
18	4,0
17	3,8
16	3,7
15	3,5
14	3,3
13	3,2
12	3,0
11	2,8
10	2,7
9	2,5
8	2,3
7	2,2
6	2,0
5	1,8
4	1,7
3	1,5
2	1,3
1	1,2
0	1,0

Vraag-identificatiecodes

Deze identifiers kunnen worden gebruikt om de precieze vraag in de vragenbanken te identificeren. Gebruik deze code in combinatie met de documentcode wanneer u feedback doorgeeft, zodat precies duidelijk is op welke vraag en -versie uw feedback van toepassing is.

Documentidentificatiecode: 31897-51652

Vraagnummer	Vraag-identificatiecode	Versie-identificatiecode
1	411302	82e3aa8a-5642-b06b-f7c0-0ffa7292dd2c
2	411157	11088b87-25be-06d1-e889-de79865a9149
3	411182	07b7adbf-826d-d4e3-3386-bc7d76a0de85
4	411222	ef21f58f-9637-1b48-4446-849cc12e9222
5	411252	c7c5cf2b-bd2e-0aad-6861-2f4201255bb0
6	411297	953fb371-bbf0-c5aa-08a0-0b12512751b2