Module: **M6. Video analysis** **Final exam**

Date: **May 6th, 2021** **Time: 2h30**

Teachers: **Montse Pardàs, Ramon Morros, Xavier Giró, Javier Ruiz, Josep Ramon Casas.**

- **Books, lecture notes, calculators, phones, etc. are not allowed.**
- **All sheets of paper should have your name.**
- **All results should be demonstrated or justified.**

Question 1. (0,75p)

Explain a possible system to detect shot transitions using a neural network. Describe the architecture and which data would be needed to train it.

See Slide Shot Segmentation XII. Each frame would be classified as either "shot transition frame" or "within shot frame". In this example 3D convolutions are used. To train the system a database can be built combining shots, so that the groundtruth can be automatically obtained.

Question 2. (1p)

Motion compensation:

a) Explain the difference between forward and backward motion compensation.
b) Why forward motion compensation results in more prediction error than backward motion compensation?

a) Forward motion estimation: All pixels in the past image are associated to a pixel in the current image (but the contrary cannot be ensured). For each position of the reference image, the motion vector is applied to find its location in the predicted frame. The value of the pixel in the reference image is copied to this location in the predicted image. Backward motion estimation: All pixels in the current image are associated to a pixel in the past image (but the contrary cannot be ensured).For each position in the predicted frame, the motion vector is applied backwards to find its reference in the reference image. The value of the pixel in the reference frame is copies to the current position.
b) Forward motion compensation results in 'holes' in the predicted image. In these holes the value of the predicted image is not defined and leads to large errors.

Question 3. (0.5p)

Optical flow: Explain the main differences between the Lucas-Kanade and Horn-Schunck methods and how these differences affect the resulting optical flow fields.

Lucas-Kanade can only provide a correct solution at some points of the image (sparse optical flow). Horn-Schunk adds a smoothing term to the functional to minimize so it can provide motion vector in all pixels (dense optical flow)

Question 4. (0.5p)

To track an object in a video sequence, the Kalman filter is used. The dynamics model of the object is perfectly accurate, with no noise, so $\boldsymbol{\sigma}_d = 0$. Describe briefly what happens in the cycles of predictions-updates.

When prediction is perfect, the measurement is not used. In this case, only the prediction steps are performed, there is no need for the correction steps.

Question 5. (0.5p)

In particle filters, explain the concept of degeneracy and how to solve it. Just give the name of the technique to solve degeneracy, it is not necessary to explain it.

Along the iterations of the particle filter algorithm, the variance of the weights increases, so that the weight of the particles tend to concentrate into a single one. This leads to a loss of diversity. To solve this problem, we can use a resampling step when degeneracy is detected.

Question 6. (0.5p)

When tracking with a particle filter, why is a particle weight normalization step needed? How is it implemented?

We ususally are able to compute the weights of the particles up to a normalization factor. As a result, the particles do not represent a true probability distribution. The normalization step divides the weight of each particle by the sum of the weights of all the particles.

Question 7. (0.5p)

Explain how Principal Component Analysis is used in Active Shape Models.

First the shape is expressed as a vector of concatenated landmark coordinates. Then these shapes are aligned with Procrustes analysis. Once aligned, PCA is applied to obtain the shape model. The shape model is defined by the most significant eigenvectors. The projection of a shape on the reduced set of eigenvectors gives the shape parameters b. Then, we can compute the plausibility of a shape for being a face according to the value of these shape parameters.

Question 8. (0.75p)

Model-based tracking:

a) When performing pose inference for model-based tracking with sampling-based methods (such as Particle Filtering), you face the problem of tracking multiple hypothesis in highly dimensional optimization space. Mention several strategies to overcome high dimensionality in stochastic optimization and briefly describe what they do

b) In the case of Layered Particle Filtering, how does the model structure generalize the different techniques above?

a) The high dimensionality optimization problem can be avoided with:

- Annealed Particle Filtering (Deutscher 2005), be spreading particles efficiently where a local minimum is more likely
- Partitioned Sampling (MacCormick 2000) and Hierarchical Particle Filtering (Bandouch 2009) by partitioning the space into a number of lower-dimensional spaces.

b) Layered Particle Filtering combines Annealed Particle Filter and Partitioned Sampling by the iterative application of Particle Filtering in sub-state space vectors. Layers can benefit from a coarse to fine optimization by exploiting the hierarchical structure of the human model. This can be implemented by, for instance, optimizing global body translation and rotation in a first layer (with a neutral pose of limbs), followed by optimizing limb rotation in a second layer, and finally optimizing limb configuration (i.e. degree of flexion) in the third. At each layer, the optimizer works freely on the current hierarchical level from the current neutral position, while refining (with finer deviations) the previous one(s).

Question 9. (0.5p)

One way CNN approaches use to detect the 2D static pose of a human in images is by using regression CNN architectures where the network outputs a location estimation. Suppose that the pose we would like to estimate is composed of keypoints: head, left hand, right hand, waist, left foot, right foot. Could you explain in detail what would be the output of the network and a possible loss function that you can use to train it?

As we have 6 keypoint the network will output 12 values, a pair of (x,y) coordinates for each keypoint to find. A possible loss function to use is MSE with the groundtruth (x,y) coordinates of the keypoints.

Question 10. (0.5p)

Another possibility to detect 2D static poses is to use heatmaps. For a similar pose detection with 6 keypoints, explain in detail what would be the output of the network and a possible loss function to use.

We will have 6 images where each pixel represents the probability of that pixel belonging to the specific keypoint. As we are based on a probabilistic frame-work a possible loss function to use is cross-entropy.

Exercise 1. (1p)

We want to use a foreground segmentation method to detect planes in an airport. The images I(x,y,t) are in graylevel:

I(i,j,t) I(i,j,t+1) I(i,j,t+2) I(i,j,t+3)

We assume the recording starts when there is no plane in the scene, and the camera is still.

a) Propose a technique to model the background of the image with one Gaussian for each pixel. Write the equations of the algorithm to create this model. Take into account that we can have gradual changes of the illumination.
b) How could you detect a plane using this background model? How would you change the modelling equations to take into account the detections?
c) After the implementation of the algorithm, we realize that the results are not too good because the camera is not completely still. Propose two solutions to improve the results, one involving a motion estimation step and one which does not.

a) The model needs to be adaptive. Mean and variance can be updated as:

$$\mu_{(i,j)}(t) = (1-\rho)\mu_{(i,j)}(t-1) + \rho I(i,j,t)$$
$$\sigma^2{}_{(i,j)}(t) = (1-\rho)\sigma^2{}_{(i,j)}(t-1) + \rho\left(I(i,j,t) - \mu_{(i,j)}(t)\right)^2$$

b) Detect foreground when

$$\left|I(i,j,t) - \mu_{(i,j)}(t-1)\right| < \alpha\sigma_{(i,j)}(t).$$

and update model according to

$$\mu_{(i,j)}(t) = \begin{cases} (1-\rho)\mu_{(i,j)}(t-1) + \rho I(i,j,t), & \text{if } (i,j) \in \text{background} \\ \mu_{(i,j)}(t-1), & \text{other} \end{cases}$$
$$\sigma^2{}_{(i,j)}(t) = \begin{cases} (1-\rho)\sigma^2{}_{(i,j)}(t-1) + \rho\left(I(i,j,t) - \mu_{(i,j)}(t)\right)^2, & \text{if } (i,j) \in \text{background} \\ \sigma^2{}_{(i,j)}(t-1), & \text{other} \end{cases}$$

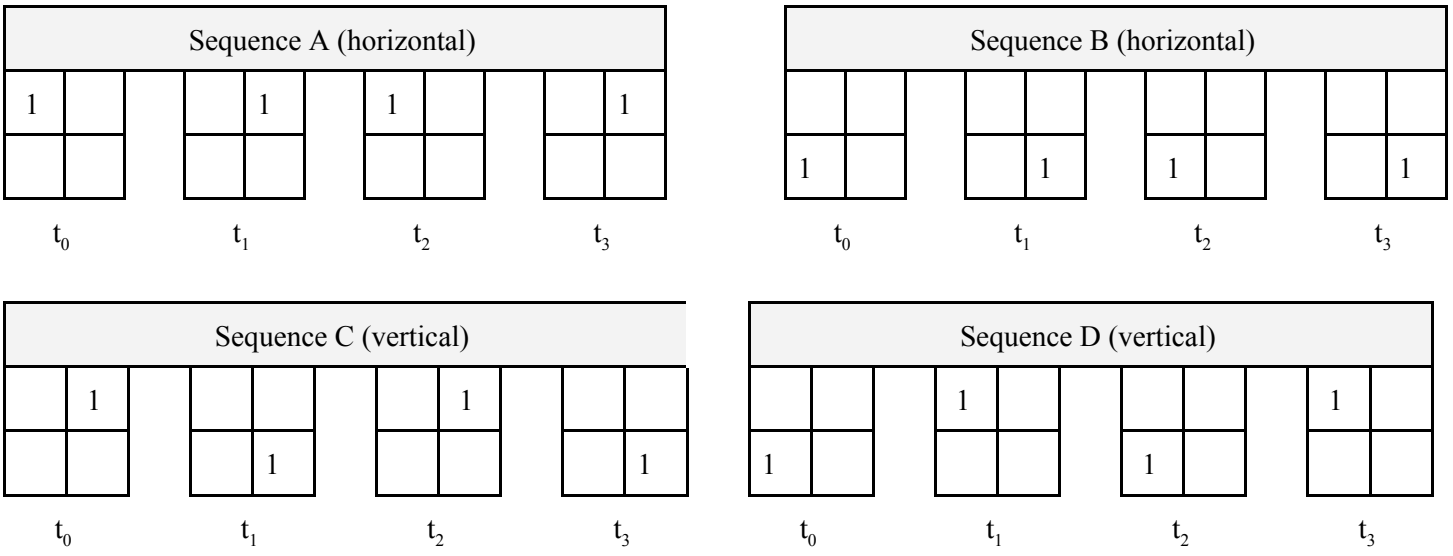
c) With motion estimation: We can select a motion model and do a global motion compensation. Then the previous technique can be applied on this result.
Without motion estimation: Assuming camera motion is small, we can assume that each pixel can only have a reduced number of values. That is, each pixel can have different stated, but the number of states is low. Then, the background can be model with a mixture of Gaussians, similar to Stauffer and Grimson method.

Exercise 2. (3p)

Consider a dataset of very short videos of 4 frames long, and 2x2 pixels of spatial definition. All videos show a 1x1 pixel over a background. The pixel may bounce in two different directions, horizontally or vertically, depending on the class of the video.

The starting coordinates of the pixel are randomly chosen within the dataset. For the purpose of this exercise, consider the following four sequences (A, B, C, D)

- Sequences A & B: Horizontally (left-right).
- Sequences C & D: Vertically (top-down).

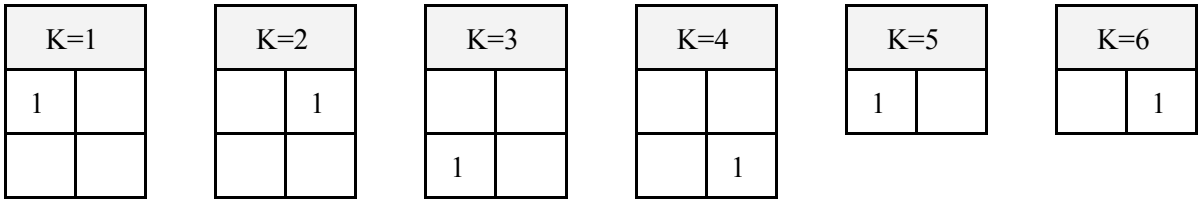


The '1' cells corresponds to the bouncing pixel, and all blank cells correspond to the background, encoded with a '0' value:

The main purpose of this problem is studying the problem by considering different neural architectures for the task of video classification between the horizontal or vertical classes.

<u>CONV-2D FEATURE EXTRACTION</u>

Study the case of a very simple feature extractor composed of a single 2D convolutional layer defined by the following four 2x2 filters, K={1,2,3,4}, in Figure 2.



<u>Figure 2</u>: Convolutional filters needed for exercises a) K={1, 2, 3, 4}, and c) K={5,6}.

a) (0,5 p) Compute the output of the four filters after the ReLU layer. Provide your answers in the corresponding t_0, t_1, t_2 and t_3 columns of Figure 3.

SINGLE FRAME MODELS

b) (0,5 p) Apply a temporal max pooling along the output features of each convolutional filter. Provide your answers in the *Pool* columns of Figure 3.

K	Sequence A (horizontal)				
	t_0	t_1	t_2	t_3	Pool
1	1		1		1
2		1		1	1
3					
4					

	Sequence B (horizontal)				
	t_0	t_1	t_2	t_3	Pool
	1		1		1
		1		1	1

K	Sequence C (vertical)				
	t_0	t_1	t_2	t_3	Pool
1					
2	1		1		1
3					
4		1		1	1

	Sequence D (vertical)				
	t_0	t_1	t_2	t_3	Pool
		1		1	1
	1		1		1

Figure 3: Provide your answers for exercises a) and b).

C(2+1D)

c) (0,5p) Instead of the temporal pooling, consider now the two 1D-temporal filters K={5,6} with stride 2, and apply them over the outputs, followed by another ReLU activation. Provide your answers in columns K={5, 6} in Figure 4.

K	Sequence A (horizontal)					
	t_0	t_1	t_2	t_3	K=5	K=6
1	1		1		1	1
2		1		1		
3						
4						

	Sequence B (horizontal)					
	t_0	t_1	t_2	t_3	K=5	K=6
	1		1		1	1
		1		1		

K	Sequence C (vertical)					
	t_0	t_1	t_2	t_3	K=5	K=6
1						
2	1		1		1	1
3						
4		1		1		

	Sequence D (vertical)					
	t_0	t_1	t_2	t_3	K=5	K=6
		1		1		
	1		1		1	1

Figure 4: Copy your answer for exercise a), and provide your answers for exercise c).

2D CNN + RNN

From now on, consider the output of the four C2D filters at timesteps t_0, t_1, t_2 and t_3 as a sequence of four tokens e_0, e_1, e_2 and e_3.

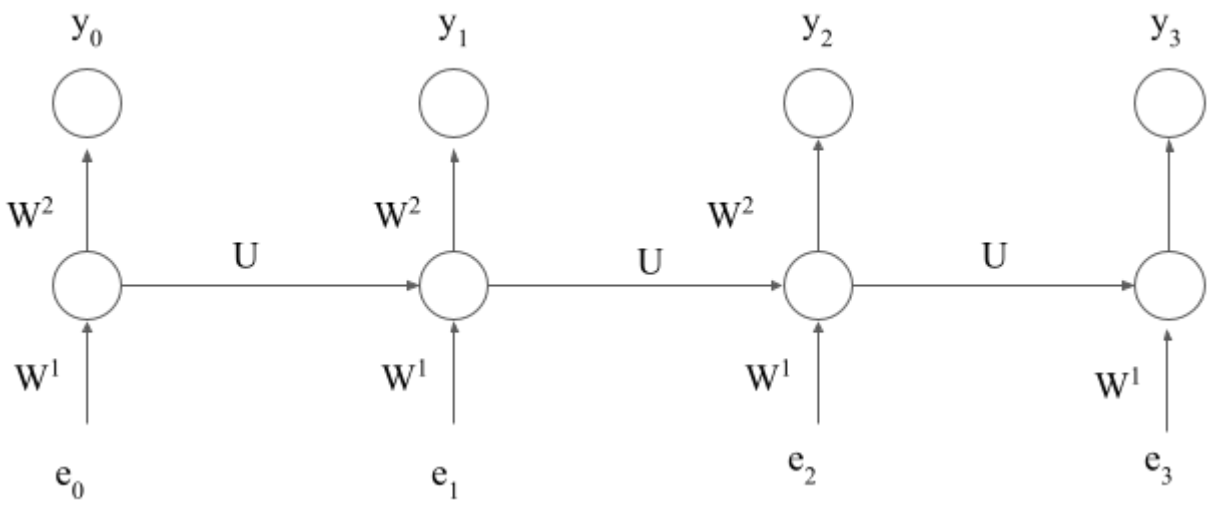
d) (0,5p) Instead of the 1D temporal convolutions in the previous section, consider now that the sequence of tokens is now fed into a single recurrent layer. If we want to keep the same amount of parameters (ignoring biases), how many neurons are there in this recurrent layer ? Develop your reasoning and calculations.

The 1D temporal convolutional layer is defined by two filters of size 2. So it contains 4 parameters.

The input feature after the 2D convolutions is of size 4.

As a result, the recurrent layer can only be composed of a single recurrent neuron.

e) (0,5p) Draw a temporally unfolded representation of the RNN architecture that depicts 4 timesteps over the sequence e_0, e_1, e_2 and e_3. Refer to the feedforward layers as W_1 (recurrent layer) and W_2 (output layer), and the weights of the recurrent layer as U. Ignore the biases in your figure.



CONTEXT-AWARE TOKENS

f) (0,5p) Draw a temporally unfolded representation of a self-attention mechanism to compute the context vector e'_2, assuming a single head of projection matrices W^Q, W^K and W^V for the queries, keys and values, respectively. Consider as inputs the set e_0, e_1, e_2 and e_3 tokens. Do not consider any positional encoding in this exercise.