Exercise 06 for MA-INF 2201 Computer Vision WS23/24
03.12.2023
Submission on 10.12.2023

1. **Theory Question:** submit solution in a separate file *task1.pdf.*

   Prove the following property of k dimensional Gaussian distributions $\text{Norm}_x[\mu, \Sigma]$:

   $$\int \text{Norm}_x[a, A]\text{Norm}_x[b, B]dx = \text{Norm}_a[b, A+B] \int \text{Norm}_x[\Sigma_*(A^{-1}a+B^{-1}b), \Sigma_*]dx,$$

   where $\Sigma_* = (A^{-1} + B^{-1})^{-1}$.

   (5 points)

2. **Programming:** submit solution in a separate file *task2.py*

   In this exercise we want to perform background subtraction for the provided image. The image comes with a rectangular bounding box that contains some skin color pixels (foreground). For this task you are required to implement a Gaussian Mixture Model and the EM algorithm for training. Assume that all covariance matrices are diagonal.

   (a) Implement the function *fit_single_gaussian* which fits a single Gaussian to provided data.
   (1 point)

   (b) GMMs rely on a good initialization. One strategy is to start with a single Gaussian model, split it into two distributions (GMM with two mixtures) and train it using the EM algorithm. For a GMM with four mixtures, both of the previous distributions can be splitted again. Implement the *split* function that doubles the number of components in the current Gaussian mixture model. In particular, generate $2K$ components out of K components as follows:
      - Duplicate the weights λ_k so you have $2K$ weights. Divide by two to ensure $\sum_k \lambda_k = 1$.
      - For each mean μ_k, generate two new means $\mu_{k1} = \mu_k + \epsilon \cdot \sigma_k$ and $\mu_{k2} = \mu_k - \epsilon \cdot \sigma_k$.
      - Duplicate the K diagonal covariance matrices so you have $2K$ diagonal covariance matrices.

   (2 points)

   (c) Implement the EM algorithm to train the GMM.
   (5 points)

   (d) Background Subtraction. Train a GMM with 8 components (start with a single Gaussian and do 3 component splits) for the background pixels. Using the thresholding approach from the lecture, set every pixel in the image to zero which is above a threshold τ. Display the resulting image.
   (2 points)