1. Let \( u : \Omega \subseteq \mathbb{R}^2 \to \mathbb{R} \) and suppose that \( u_{x_1}(x), u_{x_2}(x) \sim \mathcal{N}(0, \sigma_p) \) are iid for \( x \in \Omega \). Show that
\[
p(u) \propto \exp \left( -\frac{1}{2\sigma_p^2} \int_\Omega |\nabla u(x)|^2 \, dx \right).
\]
(Follow the arguments in Lecture 1 for computing the likelihood \( p(I|u) \)).

2. Let \( U = \{ u \in C^2(\Omega, \mathbb{R}) : \int_\Omega |\nabla u(x)|^2 \, dx < \infty, \frac{\partial u}{\partial n}(x) = 5 \text{ for } x \in \partial \Omega \} \).

Compute the space \( V \), the space of permissible perturbations of \( U \). That is, compute the largest set, \( V \), such that \( u + tv \in U \) for \( t \) small for all \( v \in V \) and each \( u \in U \). Show that \( U \neq V \), unlike the example in class.

3. Show that the Divergence Theorem
\[
\int_\Omega \text{div} \left( U(x)v(x) \right) \, dx = \int_\Omega \left( U(x) \cdot N(x) \right) v(x) \, dS(x)
\]
for \( U : \Omega \subseteq \mathbb{R}^n \to \mathbb{R}^n \) and \( v : \Omega \to \mathbb{R} \) is equivalent to integration by parts
\[
\int_\Omega U(x) \cdot \nabla v(x) \, dx + \int_\Omega \text{div} \left( U(x)v(x) \right) \, dx = \int_\Omega \left( U(x) \cdot N(x) \right) v(x) \, dS(x).
\]

4. Show that \( \Delta \) is rotationally invariant, that is, \( \Delta_x = \Delta_y \) or
\[
\sum_{i=0}^n \frac{\partial^2}{\partial x_i^2} u(x) = \sum_{i=0}^n \frac{\partial^2}{\partial y_i^2} u(x)
\]
for any \( u \in C^2(\Omega; \mathbb{R}) \) and where \( y = Rx \) and \( R \) is an \( n \times n \) rotation matrix (\( R^T R = I_{d_n} \) is the identity matrix). You may use the fact that \( \text{tr}(R^T AR) = \text{tr}(A) \) for any matrix \( A \), and \( \text{tr}(A) = \sum_{i=0}^n A_{ii} \).

5. Let \( E_1, E_2 : U \to \mathbb{R} \) be defined as
\[
E_1(u) = \int_\Omega (I(x) - u(x))^2 \, dx + \alpha \int_\Omega |\nabla u(x)| \, dx,
\]
and
\[ E_2(u) = \int_{\Omega} (I(x) - u(x))^2 \, dx + \alpha \int_{\Omega} \nabla u(x)^T M(x) \nabla u(x) \, dx, \]
where \( \alpha > 0 \), \( M(x) \) is a spatially varying \( 2 \times 2 \) matrix and
\[ \mathcal{U} = \{ u \in C^2(\Omega, \mathbb{R}) : \int_{\Omega} |\nabla u(x)|^2 \, dx < \infty, \frac{\partial u}{\partial n}(x) = 0 \text{ for } x \in \partial \Omega \}, \]
and \( \Omega \subset \mathbb{R}^2 \). Compute and simplify the Euler-Lagrange equations of \( E_1 \) and \( E_2 \). Are \( E_1 \) and \( E_2 \) convex functionals (provide a proof)? Are there properties of \( M \) to make \( E_2 \) convex? The two energies above constitute popular denoising algorithms, the former is known as total variation (TV) denoising and the later is known as tensor diffusion denoising or anisotropic denoising. These methods are typically used to reduce the blurring across edges in an image, in contrast to the regularization term \( \int_{\Omega} |\nabla u(x)|^2 \, dx \), which blurs in all directions equally.

6. In this exercise, you are going to explore the validity of prior assumptions of true images made in the first lecture. To find a prior on images, the best natural way to do this is to sample the space of images. Note that this is not easy because there are an infinite number (indeed even uncountable) of images! But you are going to try! Go around the KAUST campus and take pictures of random scenes with your digital camera (collect at least 200 images; the more the better; try to have as varied images as possible) call these images \( \{u_i\}_{i=1}^{200} \). Compute the derivatives \( u^i_{x_1}(x), u^i_{x_2}(x) \) for every point \( x \). Compute the histogram \( h_i \) for the set of numbers \( \{u^i_{x_1}(x), u^i_{x_2}(x)\}_{x \in \Omega} \). (The MatLab command hist will be useful). Plot a few histograms \( h_1, h_{10}, h_{100} \), and then compute the average of these histograms
\[ h(r) = \frac{1}{200} \sum_{i=1}^{200} h_i(r). \]
Show the plot of \( h \). Comment on the form of \( h_i \) and \( h \). Do they resemble a Gaussian distribution? If not, does it resemble some known form? You may want to experiment with smoothing the images a little with a lowpass filter (the MatLab command filter2 will be helpful here). For this problem, you may work with other classmates; in fact I encourage you to!