16-720B Computer Vision: Homework 3

Sajal Maheshwari
sajalm@andrew.cmu.edu

October 22, 2019

1 Lucas-Kanade Tracking

1.1

The equation for the warp function in the case of translation is given by Equation 1

$$\mathcal{W}(\mathbf{x};\mathbf{p}) = \mathbf{x} + \mathbf{p} \tag{1}$$

Therefore, the derivative of $\mathcal{W}$ w.r.t. $\mathbf{p}$ is given by Equation 2

$$\frac{\partial \mathcal{W}(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}^T} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{2}$$

The value of $\mathbf{A}$ in the least squares solution is given by Equation 3

$$\mathbf{A} = \frac{\partial \mathcal{I}_{t+1}(\mathbf{x}')}{\partial \mathbf{x}'^T} \frac{\partial \mathcal{W}(\mathbf{x};\mathbf{p})}{\partial \mathbf{p}^T} \tag{3}$$

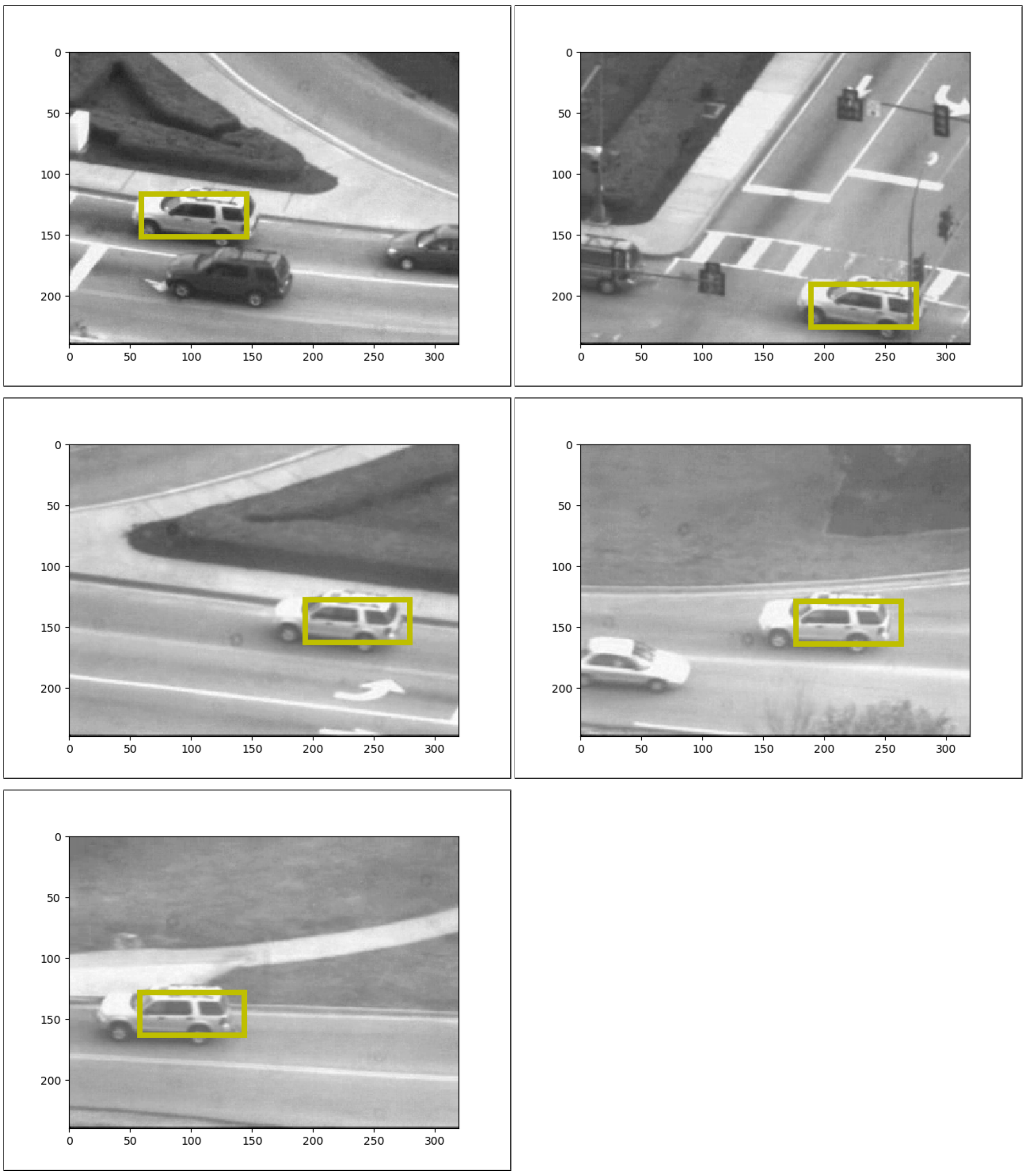
The value of $\mathbf{b}$ in the least squares solution is given by Equation 4

$$\mathbf{b} = \mathcal{I}_t(\mathbf{x}) - \mathcal{I}_{t+1}(\mathbf{x}\text{+}\mathbf{p}) \tag{4}$$

The condition on $\mathbf{A}^T\mathbf{A}$ for a unique solution of $\Delta\mathbf{p}$ is that it should not be singular. This is because if the matrix is singular, we will not be able to calculate its inverse to get a closed-form solution.

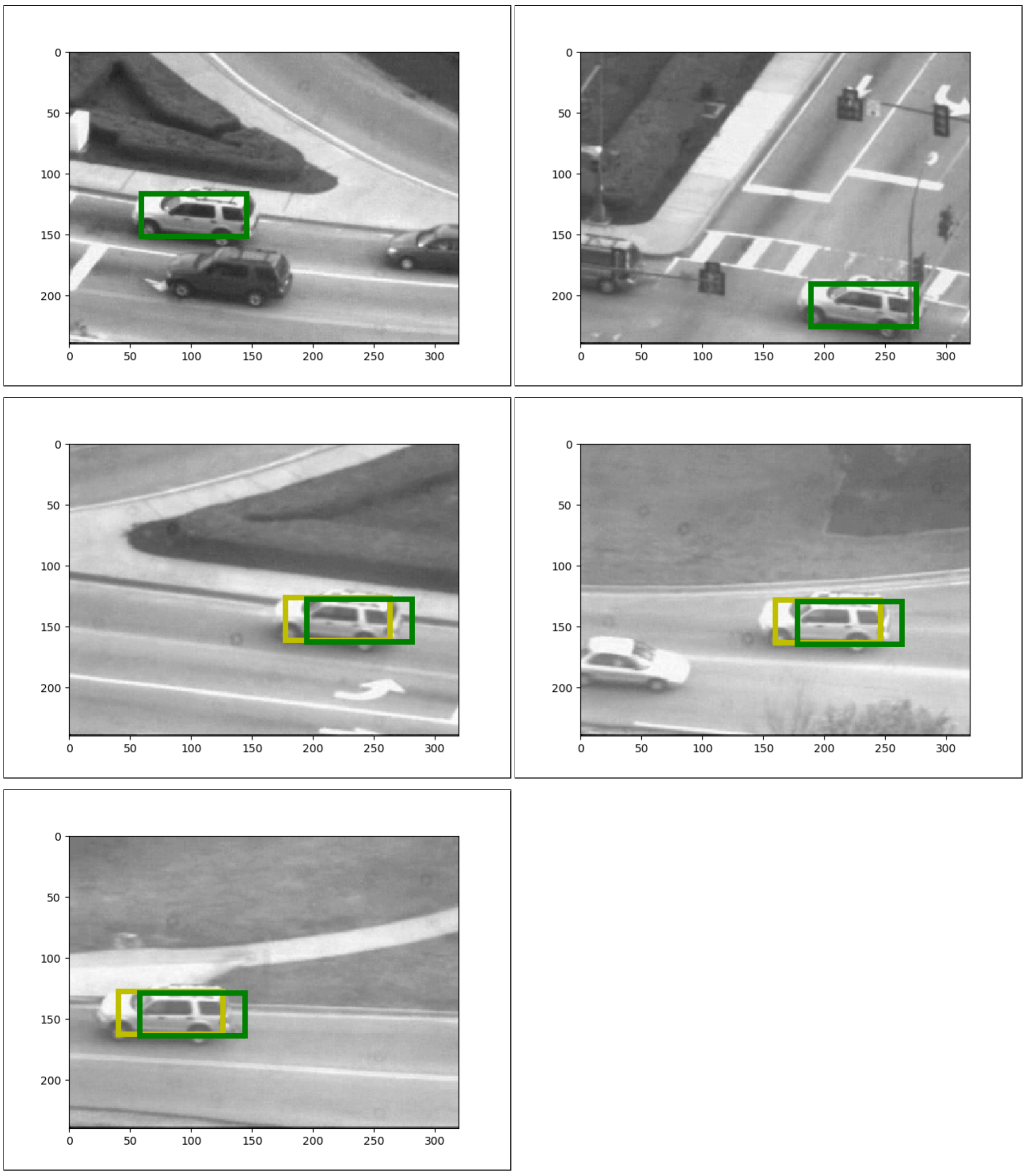
1.2

1.3



Figur 1: The figure shows examples of selected frames along with the rectangular bound which shows the output of the tracking algorithm. The frames are numbered left to right and top to bottom in the order of 1^{st}, 100^{th}, 200^{th}, 300^{th} and 400^{th} frame.

1.4



Figur 2: The figure shows examples of selected frames along with the rectangular bound which shows the output of the tracking algorithm. The frames are numbered left to right and top to bottom in the order of 1^{st}, 100^{th}, 200^{th}, 300^{th} and 400^{th} frame. The green rectangles denote the output without template correction while the yellow rectangles denote the tracking rectangles after template correction.

2 Lucas-Kanade Tracking with Appearance Basis

2.1 Appearance Basis

Traditionally in Lucas Kanade algorithm, our goal is to minimize the following equation:

$$\sum_{\mathbf{x}}[\mathcal{I}_{t+1}(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_t(\mathbf{x})]^2 \tag{5}$$

However, when we have multiple bases available which can provide a more generalized template for a particular type of data, the equation can be written as :

$$\mathcal{I}_t(\mathbf{x})+\sum_{i=1}^{m} w_i B_i(\mathbf{x}) \tag{6}$$

Therefore, we want to instead minimize the following equation :

$$\sum_{\mathbf{x}}\left[\mathcal{I}_{t+1}(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_t(\mathbf{x})-\sum_{i=1}^{m} w_i B_i(\mathbf{x})\right]^2 \tag{7}$$

If we consider the input images as vectors, the previous equation is equivalent to matrix multiplication operators, as seen in the Equation :

$$\sum_{\mathbf{x}}\left[\mathcal{I}_{t+1}(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_t(\mathbf{x})-\sum_{i=1}^{m} w_i B_i(\mathbf{x})\right]^2=\left\|\mathcal{I}_{t+1}(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_t(\mathbf{x})-\sum_{i=1}^{m} w_i B_i(\mathbf{x})\right\|^2 \tag{8}$$

Now, since we have a vectors in a subspace, if the null-space of the matrix comprising of the bases vectors exist, we can break the previous equation into the following form :

$$\left\|\mathcal{I}_{t+1}(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_t(\mathbf{x})-\sum_{i=1}^{m} w_i B_i(\mathbf{x})\right\|^2_{\mathrm{span}(B_i)}+\left\|\mathcal{I}_{t+1}(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_t(\mathbf{x})-\sum_{i=1}^{m} w_i B_i(\mathbf{x})\right\|^2_{\mathrm{span}(B_i)^1} \tag{9}$$

Now, we need to minimize :

$$\left\|\mathcal{I}_{t+1}(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_t(\mathbf{x})-\sum_{i=1}^{m} w_i B_i(\mathbf{x})\right\|^2_{\mathrm{span}(B_i)}+\|(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_t(\mathbf{x})\|^2_{\mathrm{span}(B_i)^\perp} \tag{10}$$

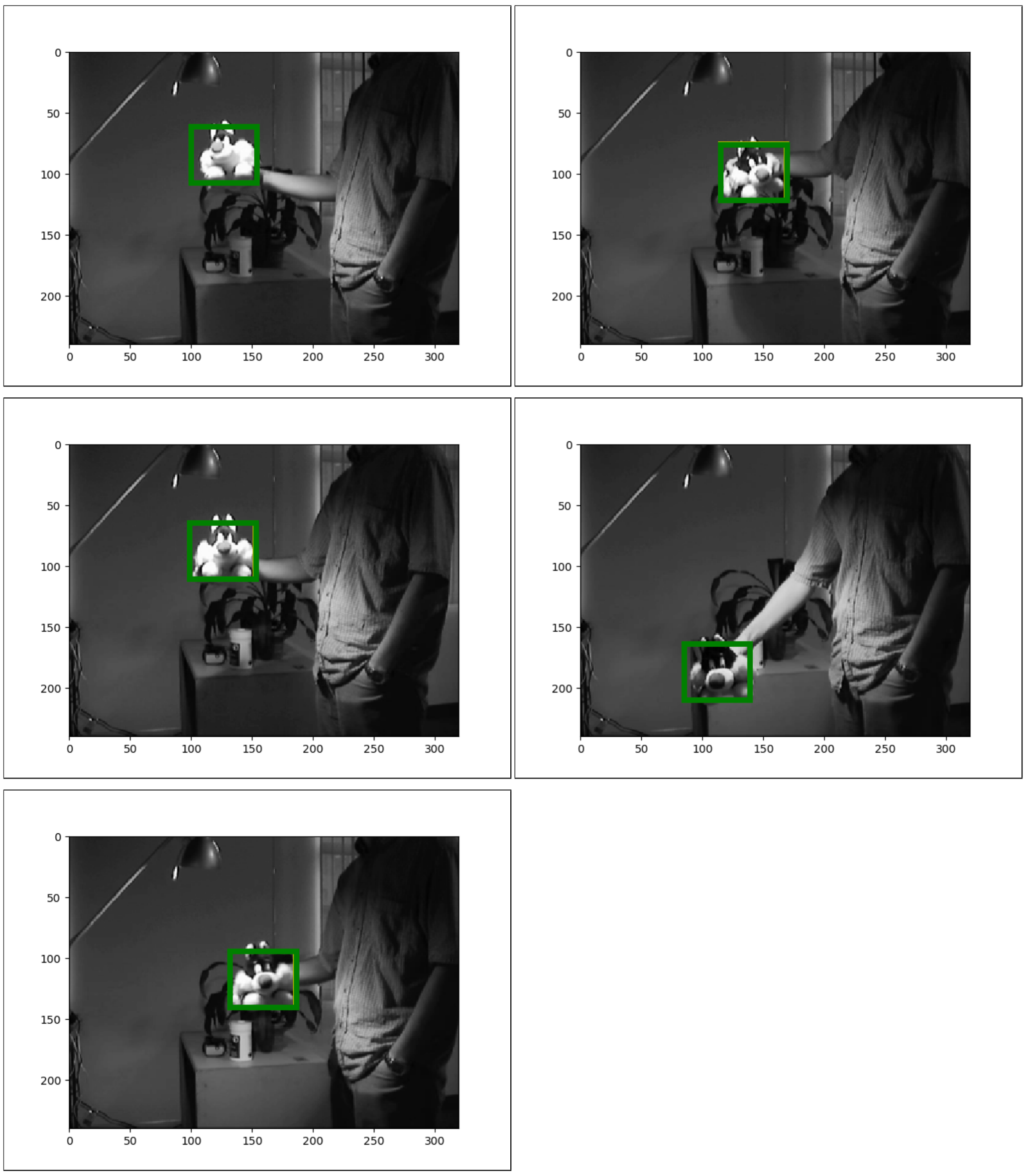
which on solving comes out to be :

$$w_i=\sum_{\mathbf{x}} B_i(\mathbf{x})\cdot[\mathcal{I}_{t+1}(\mathcal{W}(\mathbf{x};\mathbf{p}))-\mathcal{I}_t(\mathbf{x})] \tag{11}$$

However, the basis

2.2

2.3



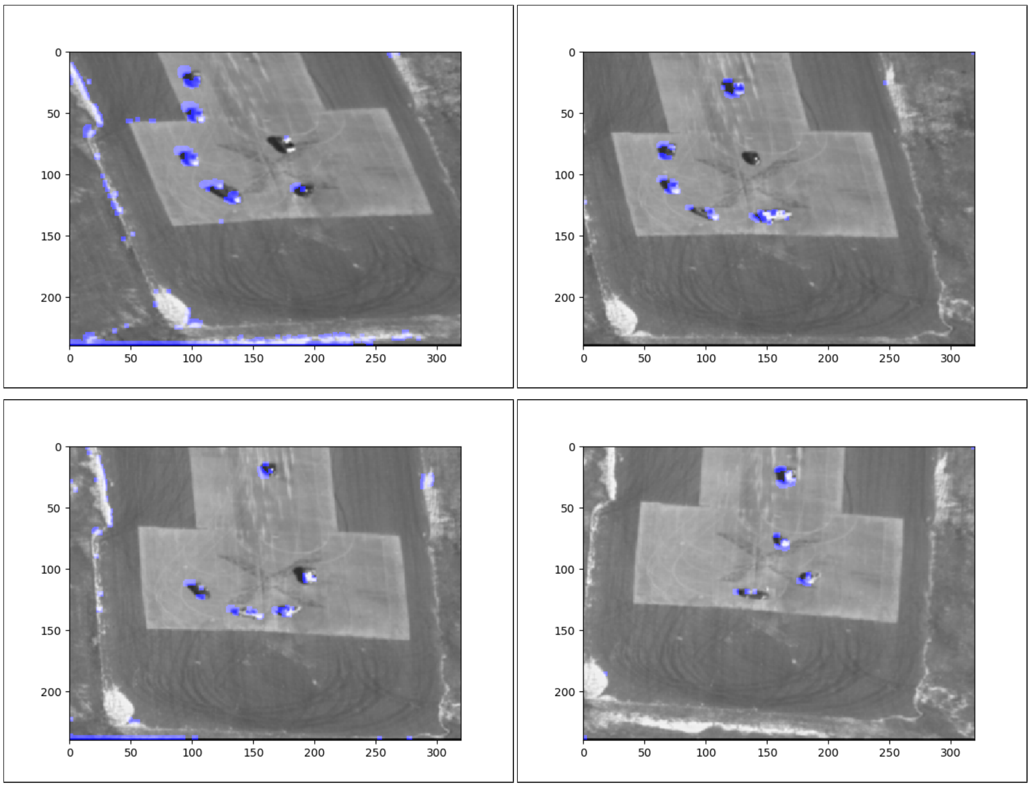
Figur 3: The figure shows examples of selected frames along with the rectangular bound which shows the output of the tracking algorithm using appearance basis. The frames are numbered left to right and top to bottom in the order of 1^{st}, 200^{th}, 300^{th}, 350^{th} and 400^{th} frame. The green rectangles denote the output without template correction while the yellow rectangles denote the tracking rectangles after template correction.

3 Affine Motion Subtraction

3.1

3.2

3.3

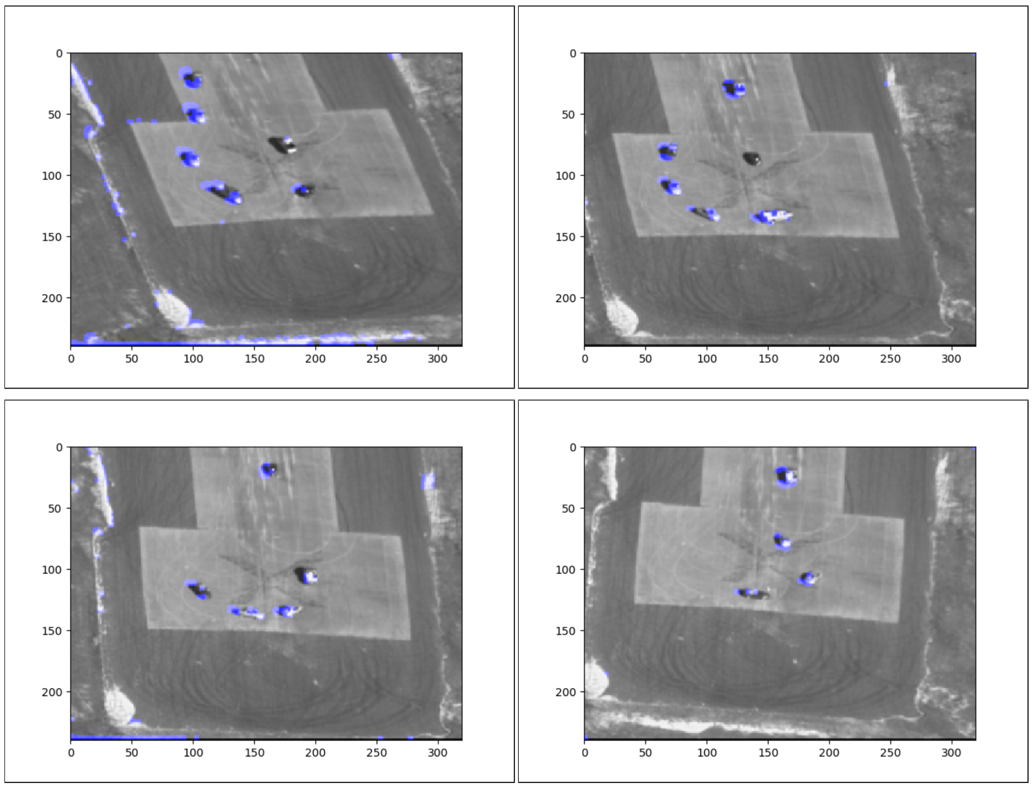


Figur 4: The figure shows examples of selected frames along with the rectangular bound which shows the optical flow output between the interest frame and the previous frame as the blue regions. The frames are numbered left to right and top to bottom in the order of 30th, 60th, 90th and 120th frame.

4 Efficient Tracking

4.1

The inverse compositional update works much faster than the traditional Lucas Kanade algorithm because the coeefficient matrix w.r.t $\Delta\mathbf{p}$ is calculated w.r.t the template image. Now, for one pair of images, the template remains constant, and therefore, this matrix is static and can be pre-computed once instead of calculation during each iteration. This results in significant reduction in the computational complexity of the solution, thus making it faster.



Figur 5: The figure shows examples of selected frames along with the rectangular bound which shows the optical flow output between the interest frame and the previous frame as the blue regions using inverse compositional update. The frames are numbered left to right and top to bottom in the order of 30^{th}, 60^{th}, 90^{th} and 120^{th} frame.

4.2

We need to solve the problem as specified in Equation 12

$$\arg\min_{\mathbf{g}} \frac{1}{2}\left\|\mathbf{y}-\mathbf{X}^T\mathbf{g}\right\|_2^2 + \frac{\lambda}{2}\|\mathbf{g}\|_2^2 \tag{12}$$

We proceed as follows :
$F = \frac{1}{2}\left\|\mathbf{y}-\mathbf{X}^T\mathbf{g}\right\|_2^2 + \frac{\lambda}{2}\|\mathbf{g}\|_2^2$
$F = \frac{1}{2}(\mathbf{y}-\mathbf{X}^T\mathbf{g})^T(\mathbf{y}-\mathbf{X}^T\mathbf{g}) + \frac{\lambda}{2}\mathbf{g}^T\mathbf{g}$
$F = \frac{1}{2}(\mathbf{y}^T-\mathbf{g}^T\mathbf{X})(\mathbf{y}-\mathbf{X}^T\mathbf{g}) + \frac{\lambda}{2}\mathbf{g}^T\mathbf{g}$
$F = \frac{1}{2}(\mathbf{y}^T\mathbf{y}-2\mathbf{g}^T\mathbf{X}\mathbf{y}+\mathbf{g}^T\mathbf{X}\mathbf{X}^T\mathbf{g}) + \frac{\lambda}{2}\mathbf{g}^T\mathbf{g}$

We need to minimize the expression above w.r.t $\mathbf{g}$. Therefore we take the derivative of F w.r.t $\mathbf{g}$ today. Setting $\frac{\partial F}{\partial \mathbf{g}} = 0$.
$\frac{\partial F}{\partial \mathbf{g}} = -\mathbf{X}\mathbf{y} + (\mathbf{X}\mathbf{X}^T + \lambda)\mathbf{g}$.
This implies :
$0 = -\mathbf{X}\mathbf{y} + (\mathbf{X}\mathbf{X}^T + \lambda)\mathbf{g}$

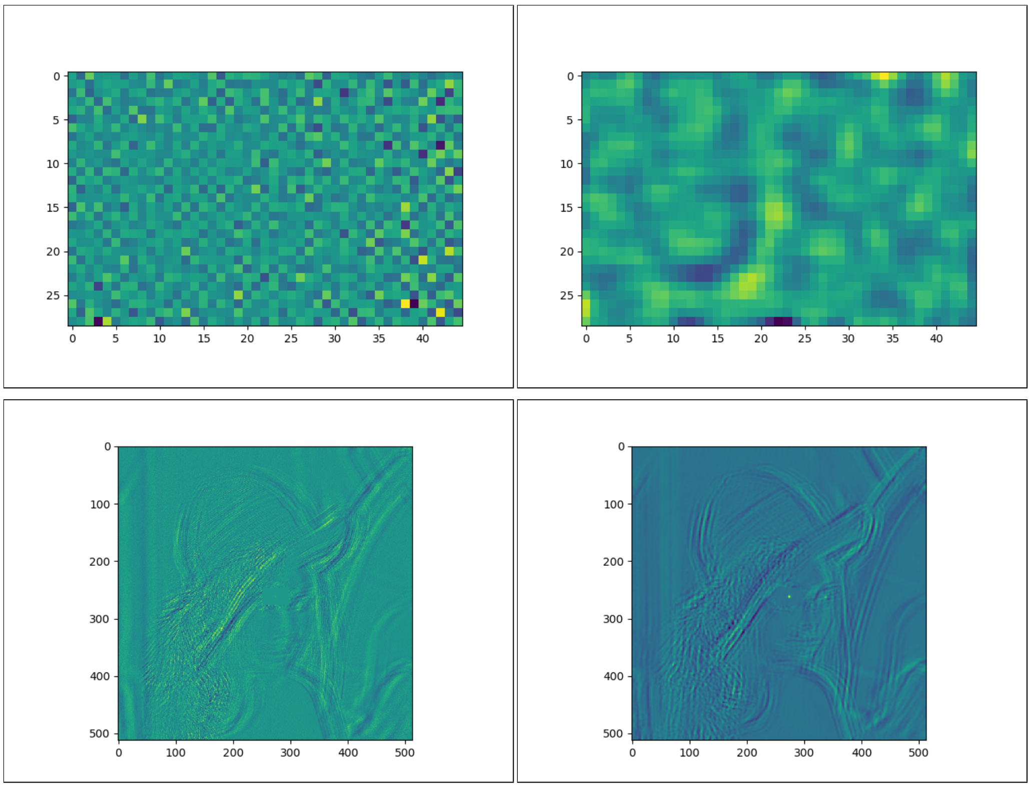
$$g = (\mathbf{X}\mathbf{X}^T + \lambda I)^{-1}\mathbf{X}\mathbf{y} \tag{13}$$

which equals :

$$g = (\mathbf{S} + \lambda I)^{-1}\mathbf{X}\mathbf{y} \tag{14}$$

4.3

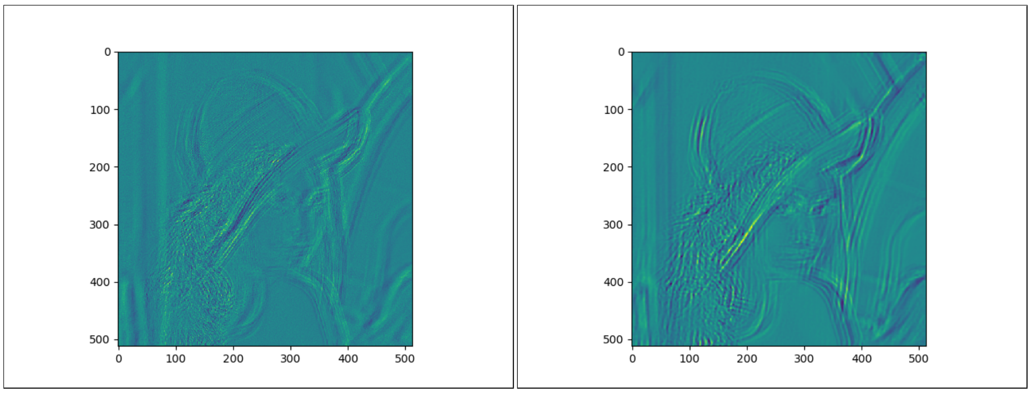
The figure shows the linear discriminant weight vectors along with the corresponding correlation outputs. The output is better for λ set to 1 because in the optimization involved in finding the correlation filters, a regularizer helps us to remain in/close to the trust region, which is desirable for better output.



Figur 6: The figure shows the linear discriminant weight vectors in the top row for the values of λ set to 0(left) and 1(right). The figure also shows the corresponding correlation outputs in the bottom row.

4.4

The convolution outputs for the weight functions are shows in Figure. Since the filter is flipped both sideways and upside down in convolution, we do not get the same output that we gor in the correlation operation. This flipping effect can be achieved in Numpy using g = g[::-1, ::-1].



Figur 7: The figure shows the convolution outputs generated by the weight function generated using λ as 0 (left) and 1(right).