# Homework 3: Md Mahin (1900421)

May 1, 2020

## 1 Solving optimization problems with CVX

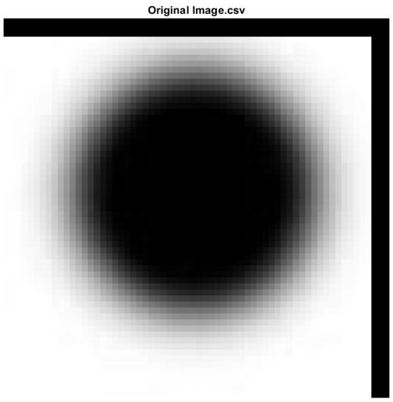
(a) Using CVX, we will solve the 2d lasso problem and its variants:

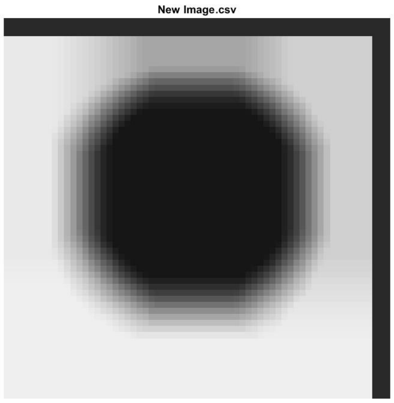
$$\min_{\theta \in \mathbb{R}^{mn}} \frac{1}{2} \sum_{i=1}^{mn} (y_i - \theta_i)^2 + \lambda \sum_{(i,j)\in E} |\theta_i - \theta_j| \, .$$

The set $E$ is the set of all undirected edges connecting horizontally or vertically neighboring pixels in the image. More specifically, $(i, j) \in E$ if and only if pixel $i$ is the immediate neighbor of pixel $j$ on the left, right, above or below.

1. Load the basic test data from `toy.csv` and solve the 2d lasso problems with $\lambda = 1$. Report the objective value obtained at the solution and plot the solution and original data as images. Why does the shape change its form?

   **Answer:**
   The objective value I got is : +116.397. The change of shape is due to two reason. First if we use 2-norm in place of 1-norm, the shape become more round. On the other hand if we reduce the value for $\lambda$ it become more round. So, high value of $\lambda$ is another reason.
   **Result:**





2. Another way to formulate the 2d lasso problem is as follows:

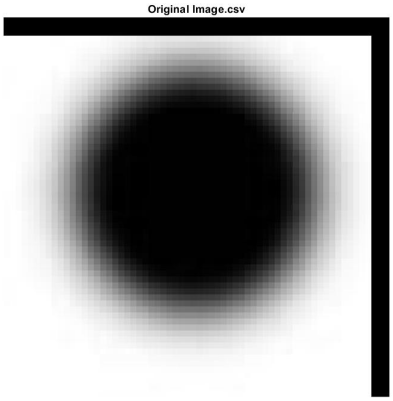
$$\min_{\theta \in \mathbb{R}^{m \times n}} \frac{1}{2} \sum_{a=1}^{m} \sum_{b=1}^{n} (y_{a,b} - \theta_{a,b})^2 + \lambda \sum_{a=1}^{m} \sum_{b=1}^{n} \left\| \begin{pmatrix} \theta_{a,b} - \theta_{a+1,b} \\ \theta_{a,b} - \theta_{a,b+1} \end{pmatrix} \right\|_p .$$

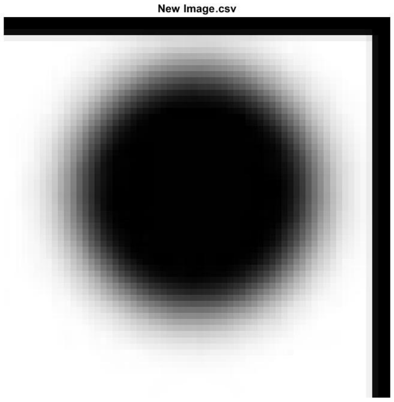
Note that the index $a, b$ here refers to the coordinates of pixel $i$. When taking a 1-norm ($p = 1$), the formulation reduces to the 2d fused lasso mentioned above, and the latter term is called an "anisotropic" total variation penalty. When taking a 2-norm ($p = 2$), the term is called an "isotropic" total variation penalty.

Solve the "isotropic" 2d lasso problems with $\lambda = 1$ on `toy.csv`. Report the objective value obtained at the solution and plot the solution and original data as images. Informally speaking, why is the output different from the "anisotropic" penalty, and what's the difference?

Hint: For `cvxpy` users, the `diff` function, the `hstack` function, and the axis option in the `norm` function would be useful. For Matlab `CVX` users, there is a `norms(x,p,dim)` function that can compute the norm along different dimensions.

**Answer:**
The objectieve value here is +7.51061. Here we use 2-norm in place of 1-norm, that is why shape change took place.
**Result:**





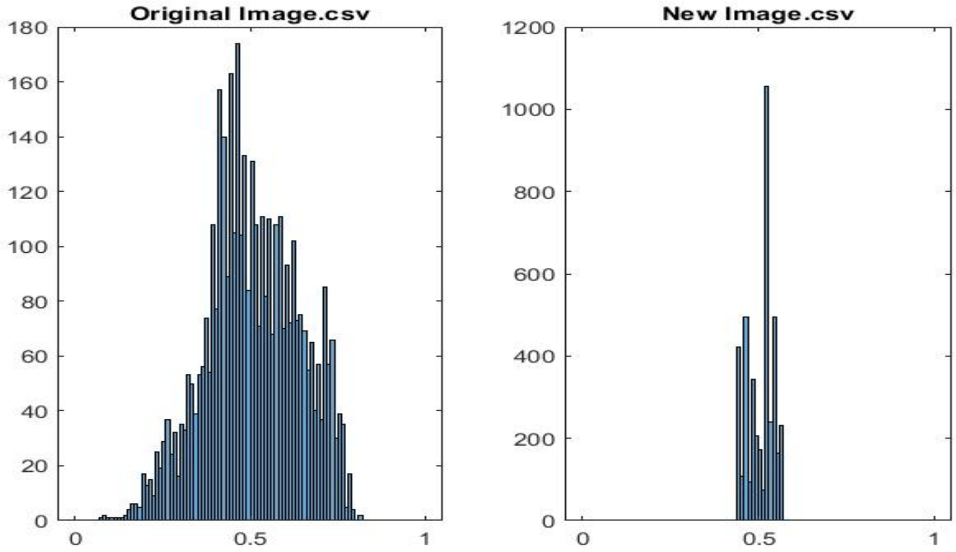
3. Next, we consider how the solution changes as we vary $\lambda$. Load a grayscale $64 \times 64$ pixel image from `baboon.csv` and solve the isotropic and anisotropic 2d lasso problem for this image for $\lambda \in \{10^{-k/4} : k = 0, 1, \ldots, 8\}$. For each $\lambda$, report the value of the optimal objective value, plot the optimal image and show a histogram of the pixel values (100 bins between values 0 and 1). What change in the histograms can you observe with varying $\lambda$ for the isotropic and anisotropic penalties?

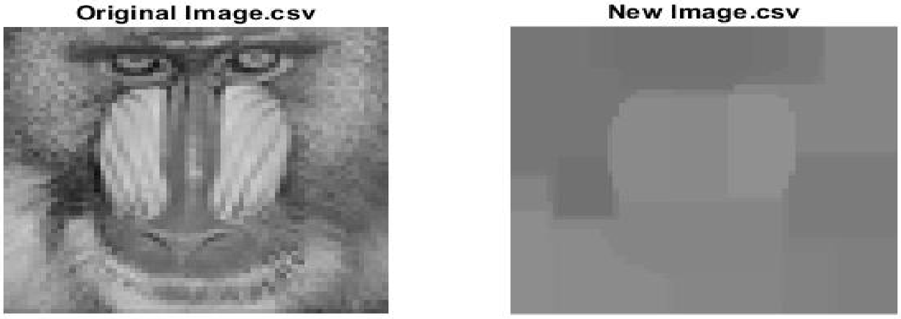
**Anisotropic**
Here, when,
k=0, $\lambda = \{10^{0/4}\}$, Optimal Value: 33.9011
k=1, $\lambda = \{10^{-1/4}\}$, Optimal Value: 27.864

k=2, $\lambda = \{10^{-2/4}\}$, Optimal Value: 21.30005
k=3, $\lambda = \{10^{-3/4}\}$, Optimal Value: 15.5876
k=4, $\lambda = \{10^{-4/4}\}$, Optimal Value: 11.0726
k=5, $\lambda = \{10^{-5/4}\}$, Optimal Value: 7.5598
k=6, $\lambda = \{10^{-6/4}\}$, Optimal Value: 4.9616
k=7, $\lambda = \{10^{-7/4}\}$, Optimal Value: 3.1267
k=8, $\lambda = \{10^{-8/4}\}$, Optimal Value: 1.8992

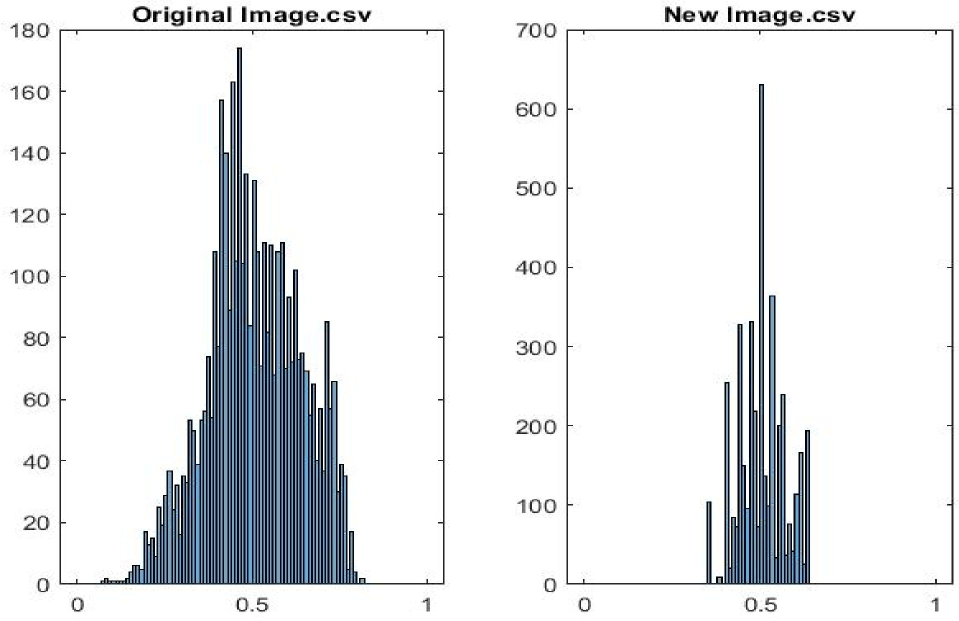
Here, We can see when the value of k is low the number of bins with values in histogram is also low. As the k value increases they starts more resembling histogram from the original image. The generated image also blurry for low k value but quality improves as k increases.

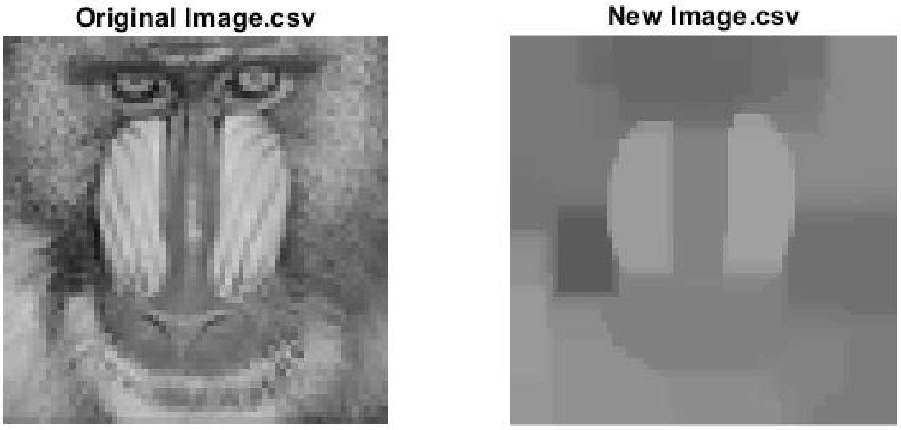
k=0,



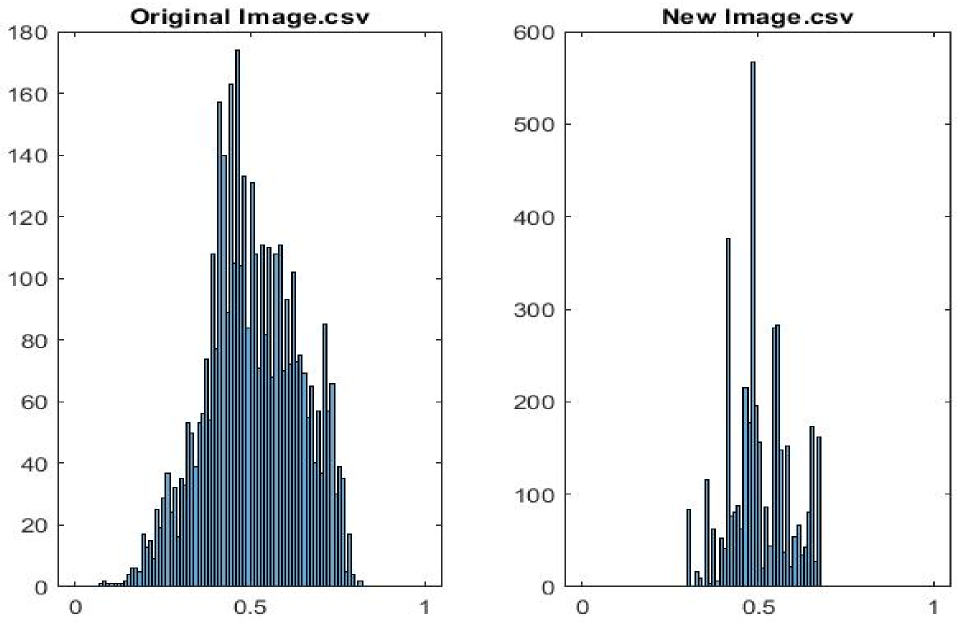


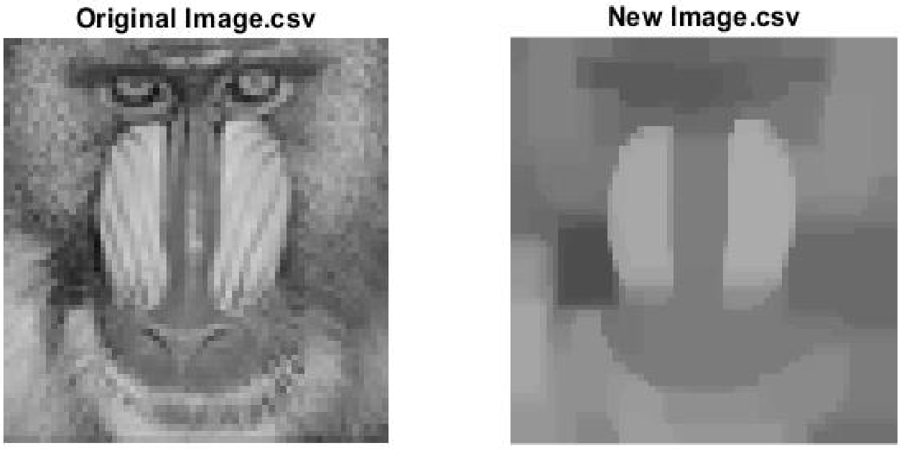
k=1


Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
700
600
500
400
300
200
100
0
0
0.5
1

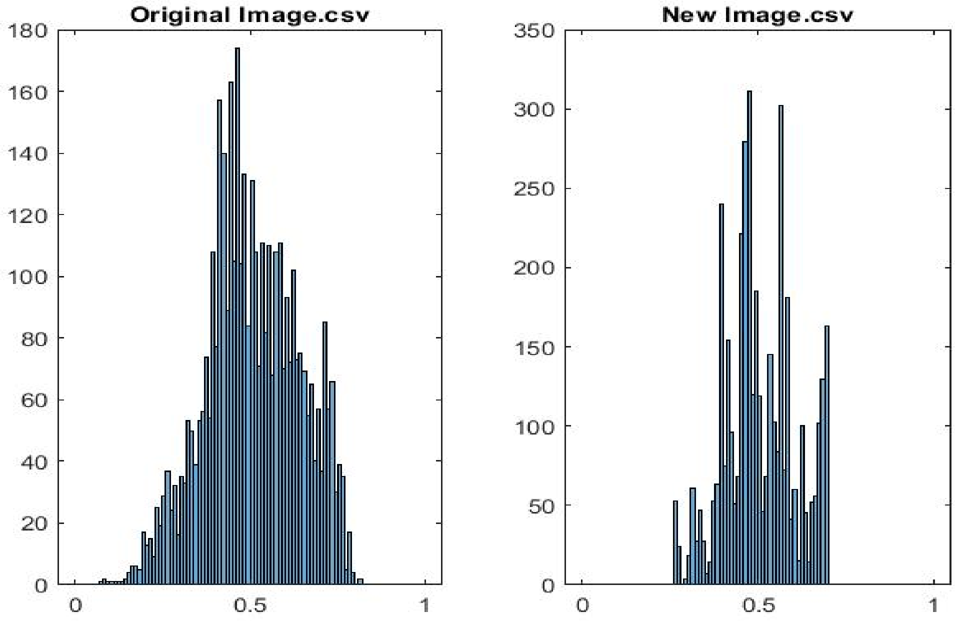

Original Image.csv
New Image.csv

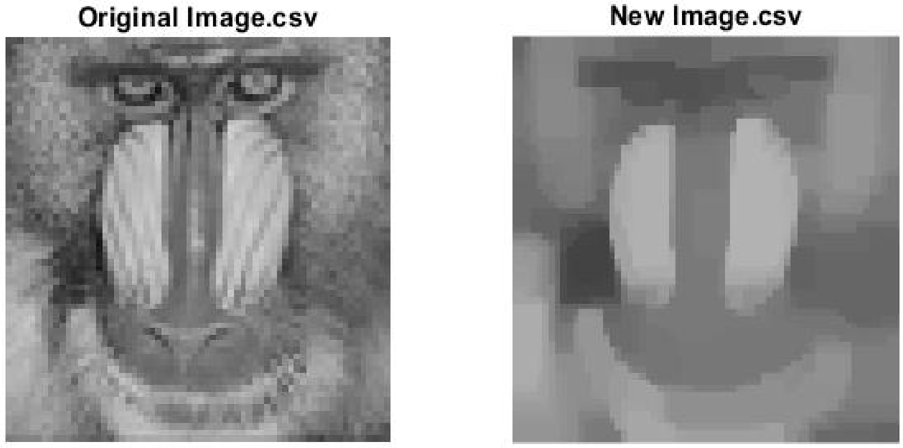
k=2


Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
600
500
400
300
200
100
0
0
0.5
1

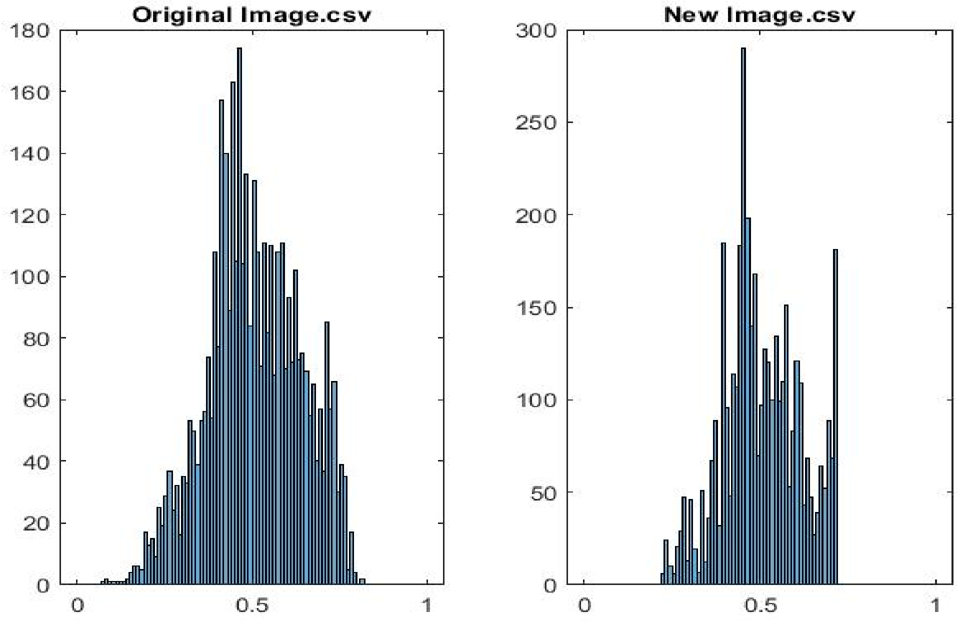

Original Image.csv
New Image.csv

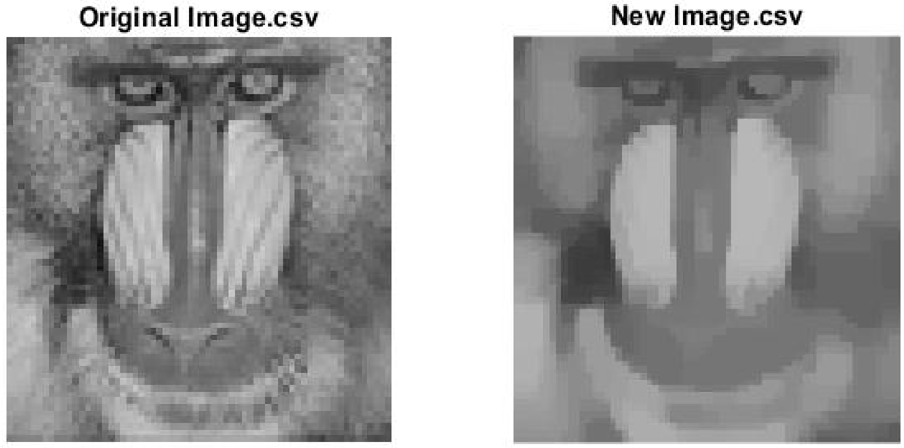
k=3


Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
350
300
250
200
150
100
50
0
0
0.5
1

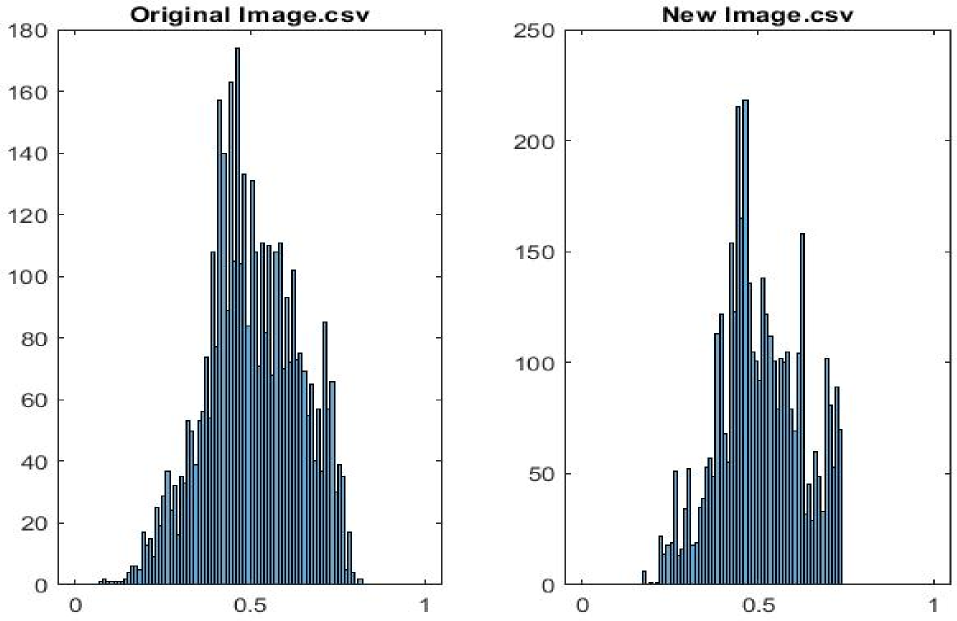

Original Image.csv
New Image.csv

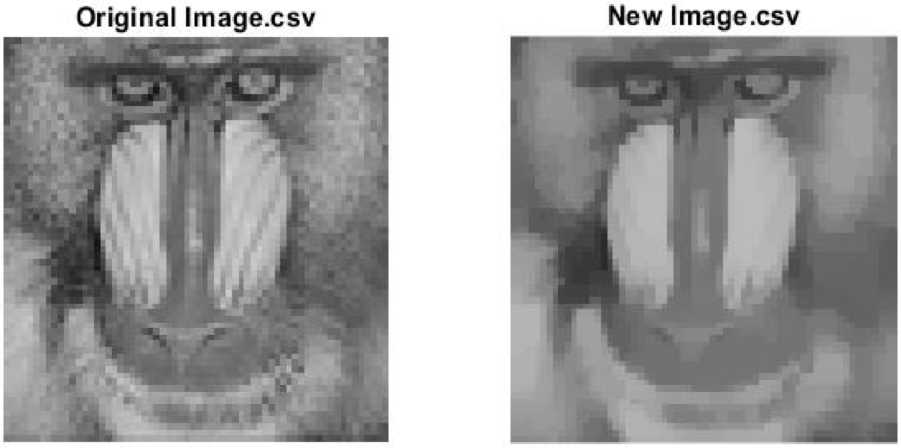
k=4


Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
300
250
200
150
100
50
0
0
0.5
1

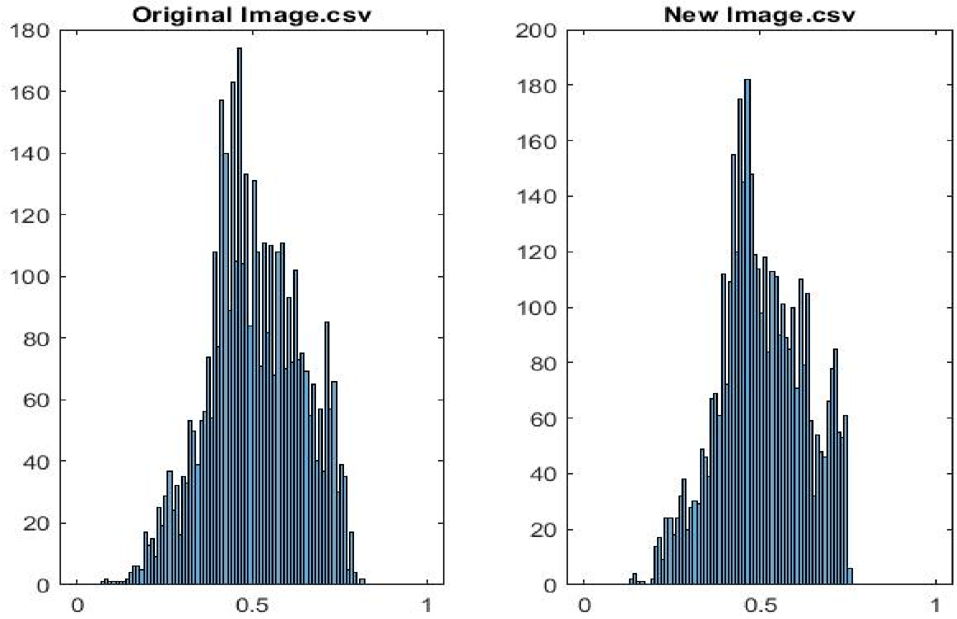

Original Image.csv
New Image.csv

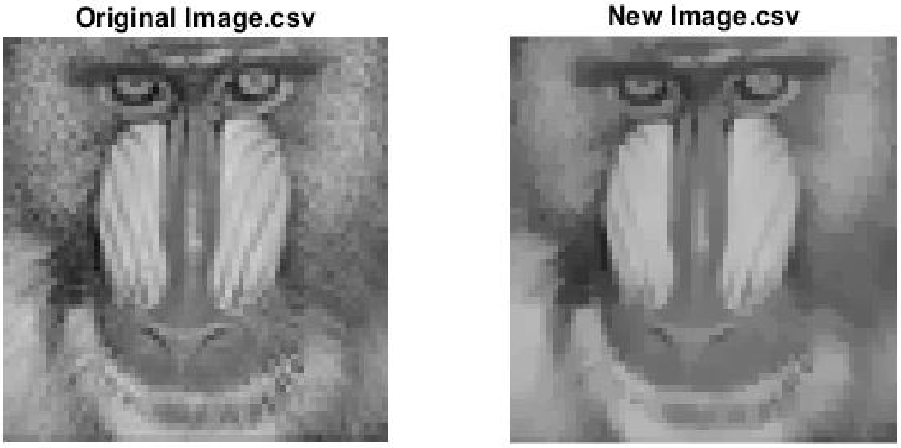
k=5


Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
250
200
150
100
50
0
0
0.5
1

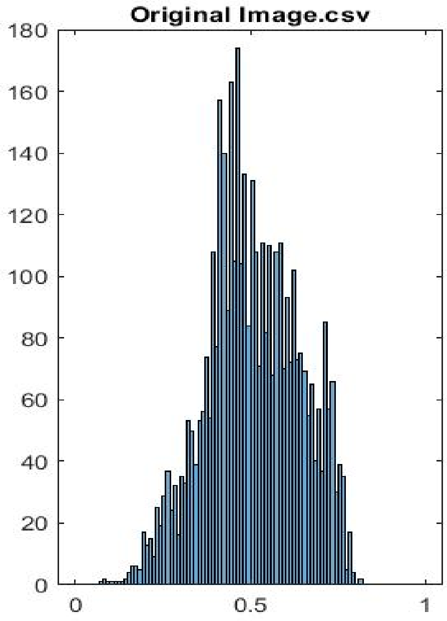

Original Image.csv
New Image.csv

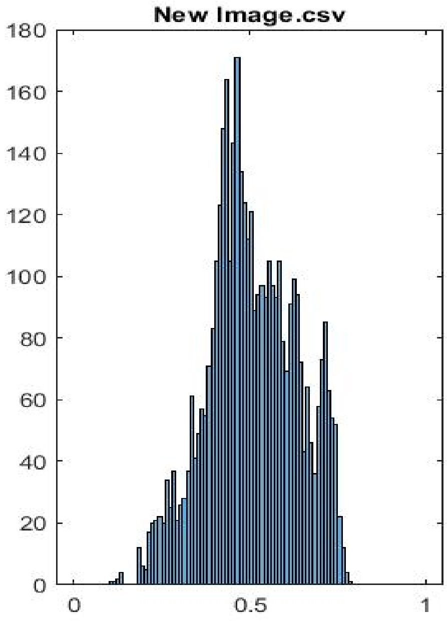
k=6

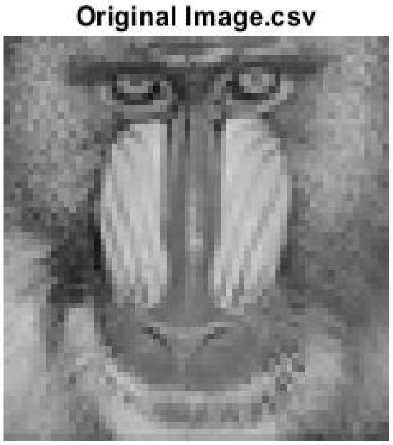

Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
200
180
160
140
120
100
80
60
40
20
0
0
0.5
1

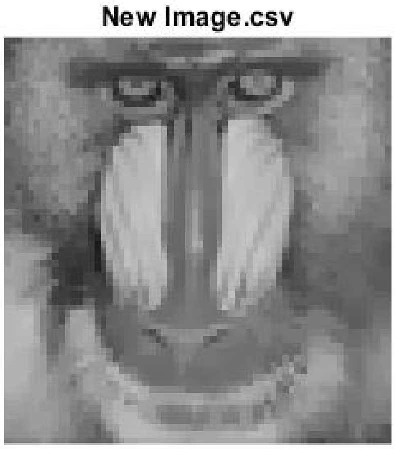

Original Image.csv
New Image.csv

k=7

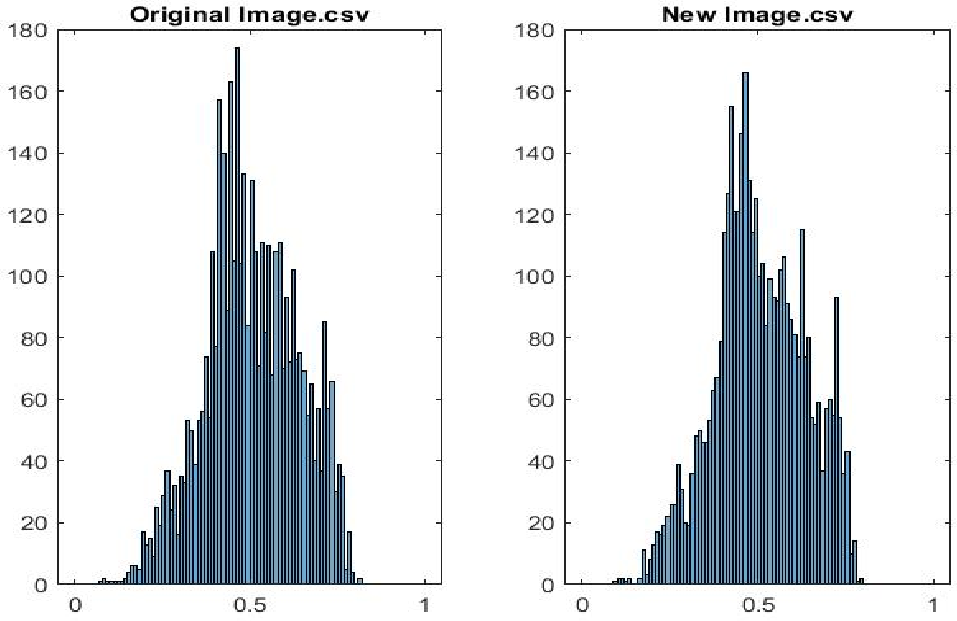

Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1

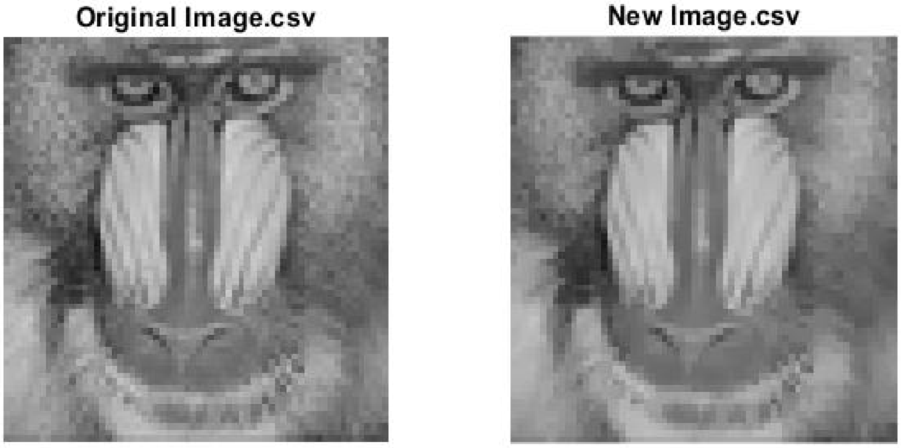

New Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1


Original Image.csv


New Image.csv

k=8

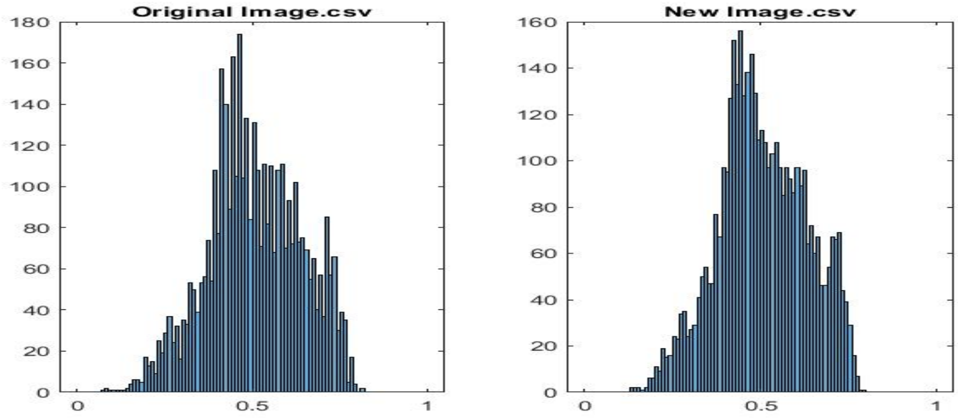

Original Image.csv
New Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1

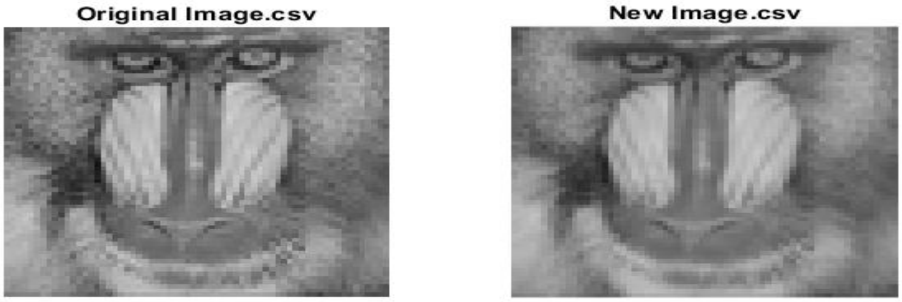

Original Image.csv
New Image.csv

**Isotropic**

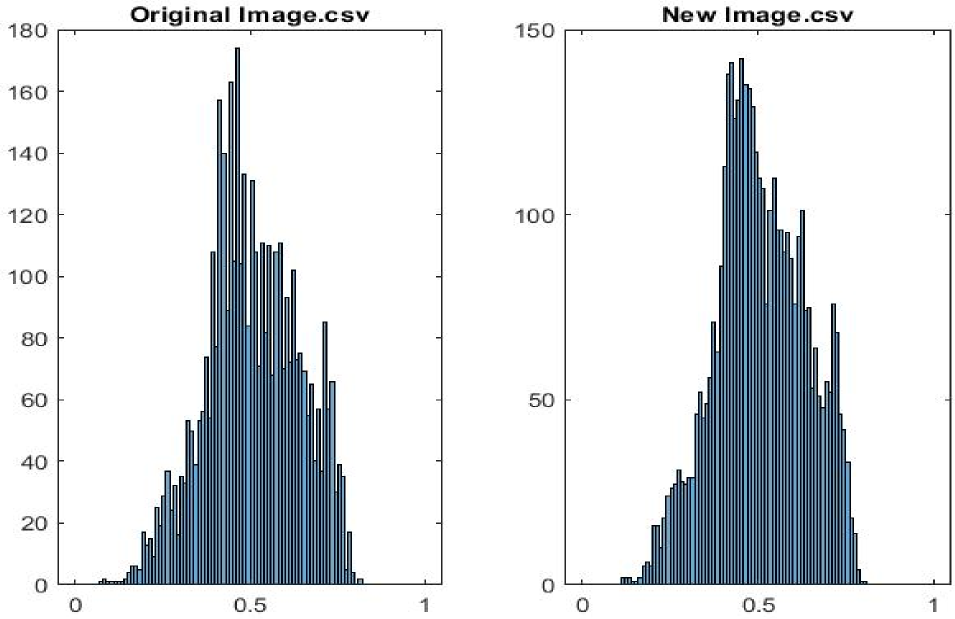
Here, when,
k=0, $\lambda = \{10^{0/4}\}$, Optimal Value: 4.093
k=1, $\lambda = \{10^{-1/4}\}$, Optimal Value: 2.4511
k=2, $\lambda = \{10^{-2/4}\}$, Optimal Value: 1.4302
k=3, $\lambda = \{10^{-3/4}\}$, Optimal Value: 0.82149
k=4, $\lambda = \{10^{-4/4}\}$, Optimal Value: 0.46757
k=5, $\lambda = \{10^{-5/4}\}$, Optimal Value: 0.26473
k=6, $\lambda = \{10^{-6/4}\}$, Optimal Value: 0.14945
k=7, $\lambda = \{10^{-7/4}\}$, Optimal Value: 0.084222
k=8, $\lambda = \{10^{-8/4}\}$, Optimal Value: 0.047419

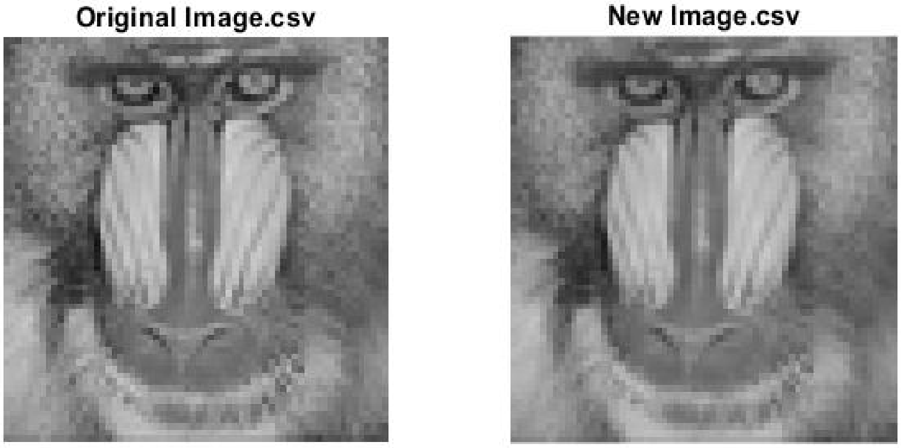
Here, We can see even with the low k, histograms are quite same. However, for low k value, from the middle point to last it is .5 to 1 seems to slightly higher value for low k and as k increases it becomes more alike to main data histogram.
k=0



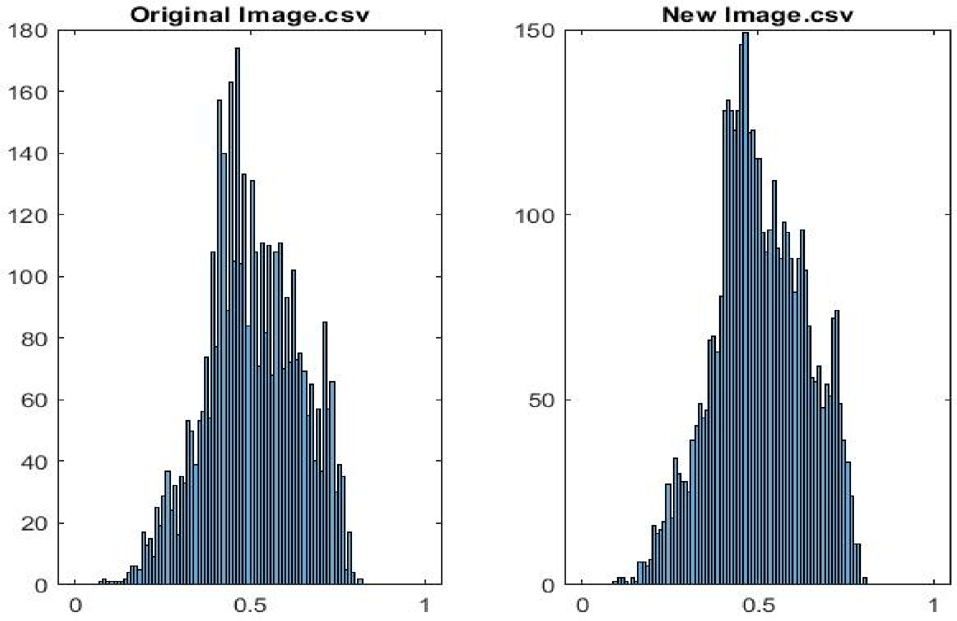


k=1

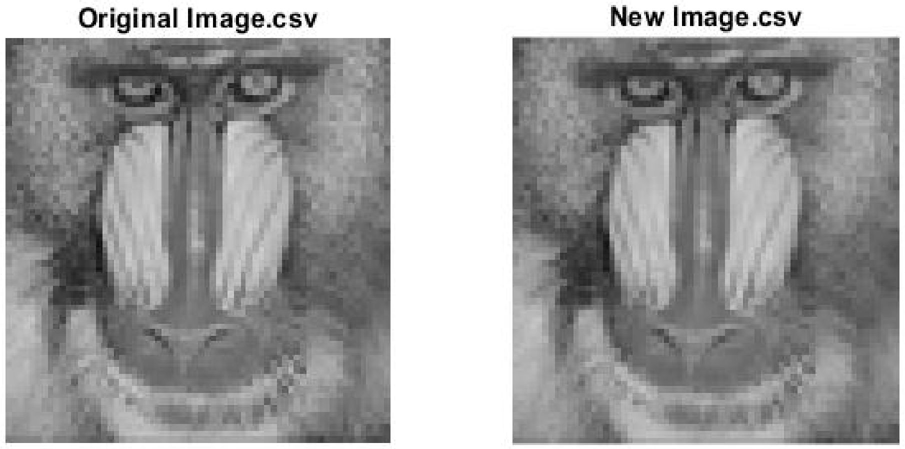

Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
150
100
50
0
0
0.5
1


Original Image.csv
New Image.csv

k=2

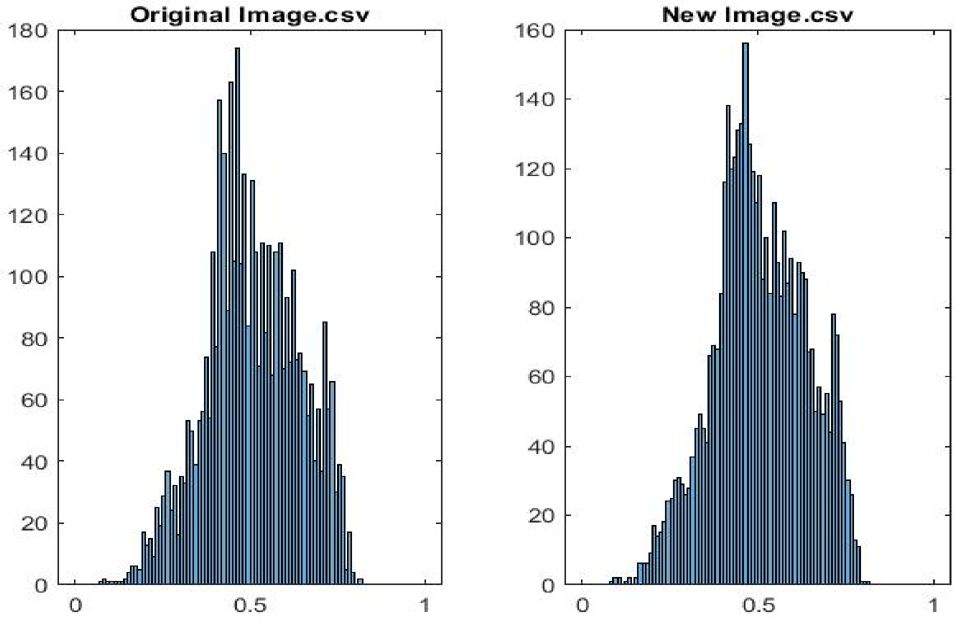


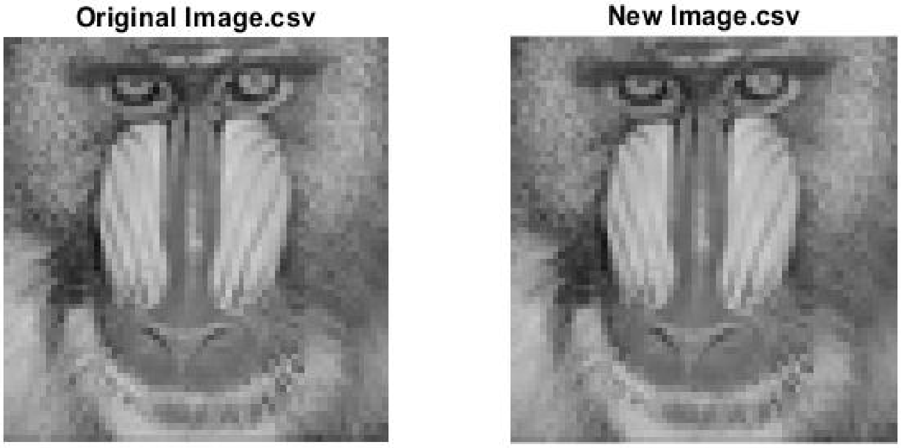
Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
150
100
50
0
0
0.5
1



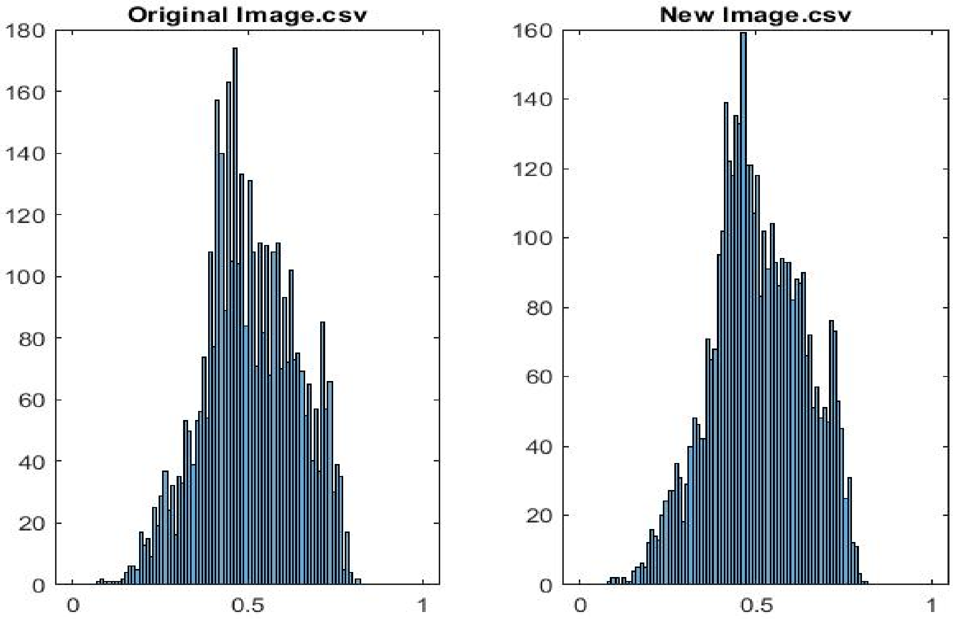
Original Image.csv
New Image.csv

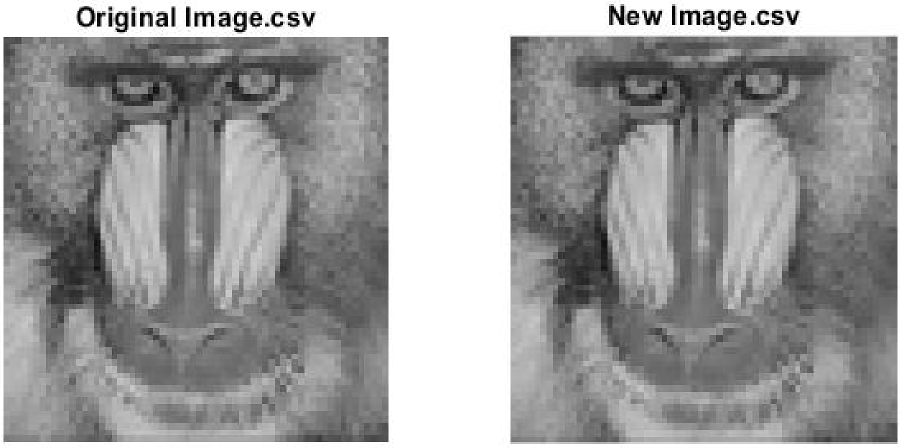
k=3


Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
160
140
120
100
80
60
40
20
0
0
0.5
1

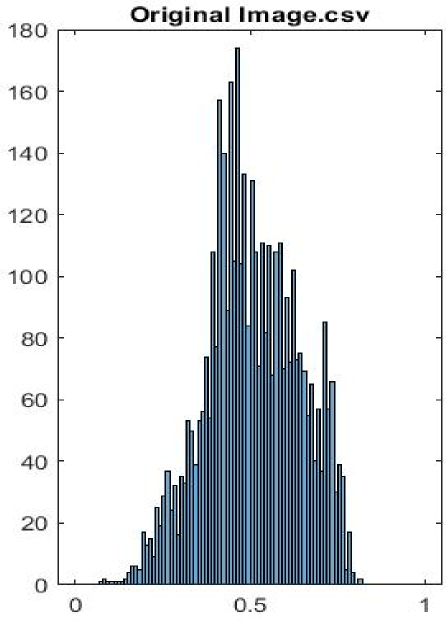

Original Image.csv
New Image.csv

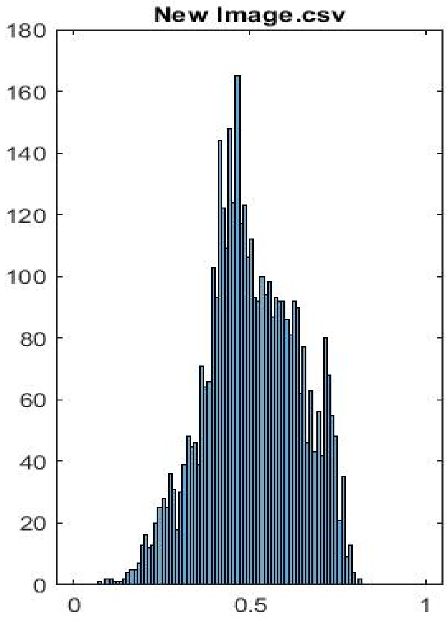
k=4

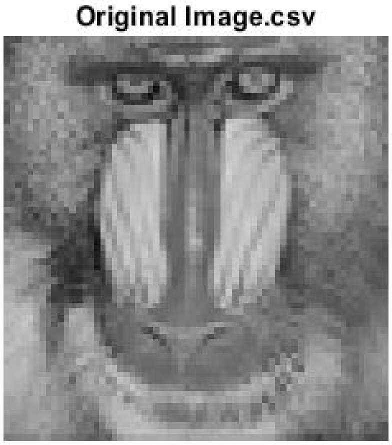

Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
160
140
120
100
80
60
40
20
0
0
0.5
1

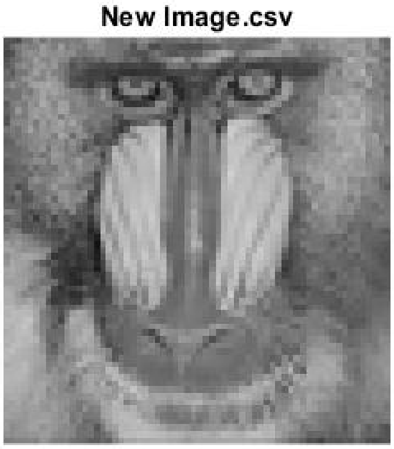

Original Image.csv
New Image.csv

k=5

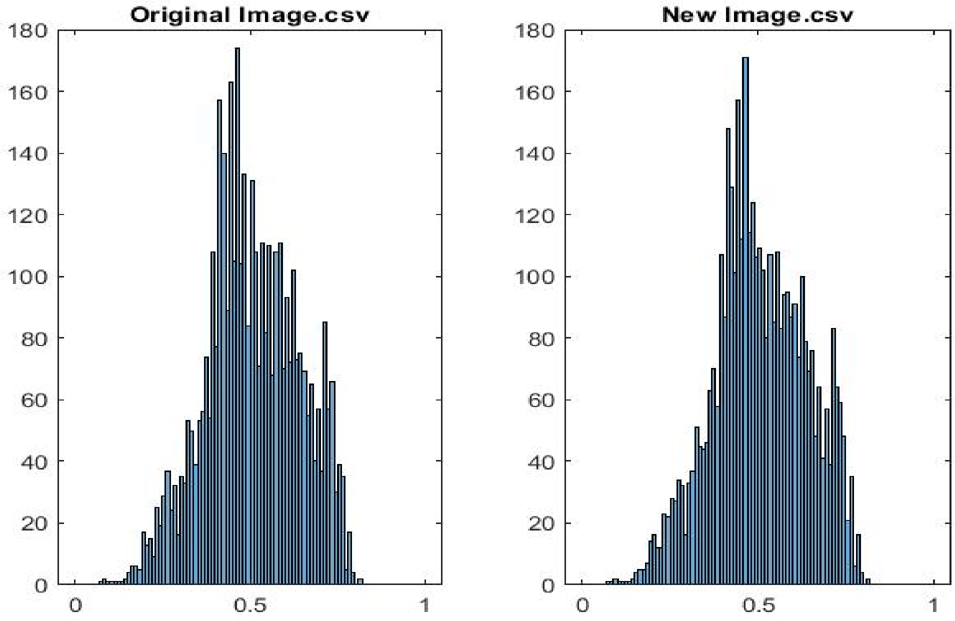

Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1

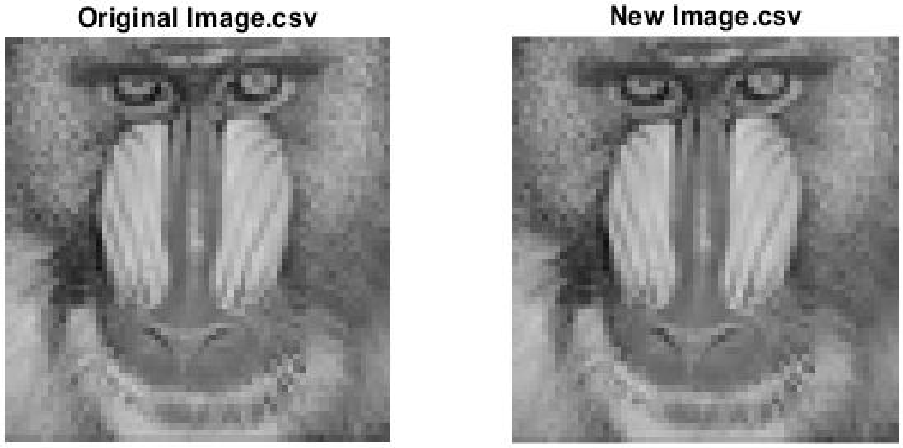

New Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1


Original Image.csv

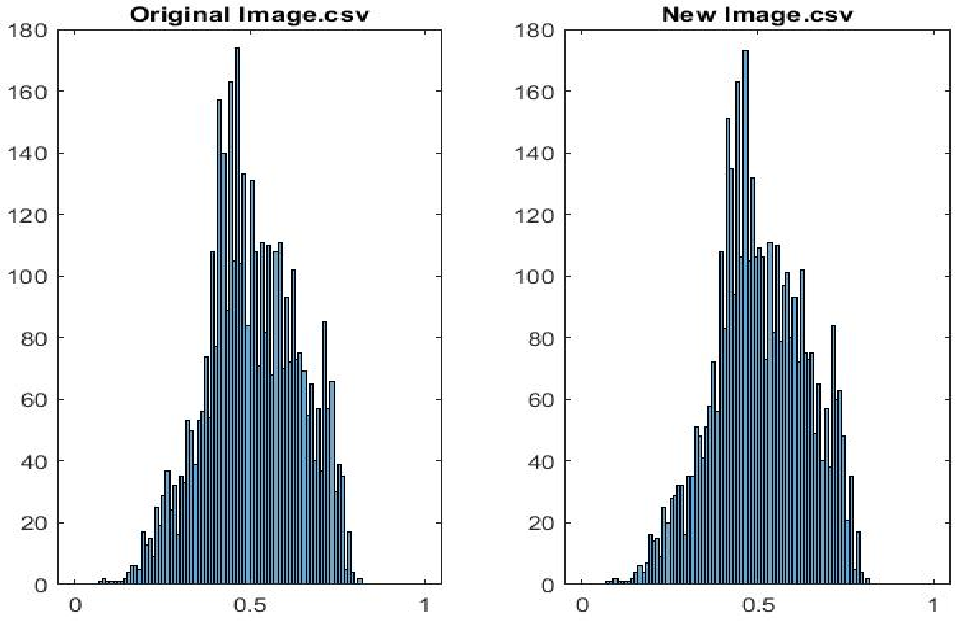

New Image.csv

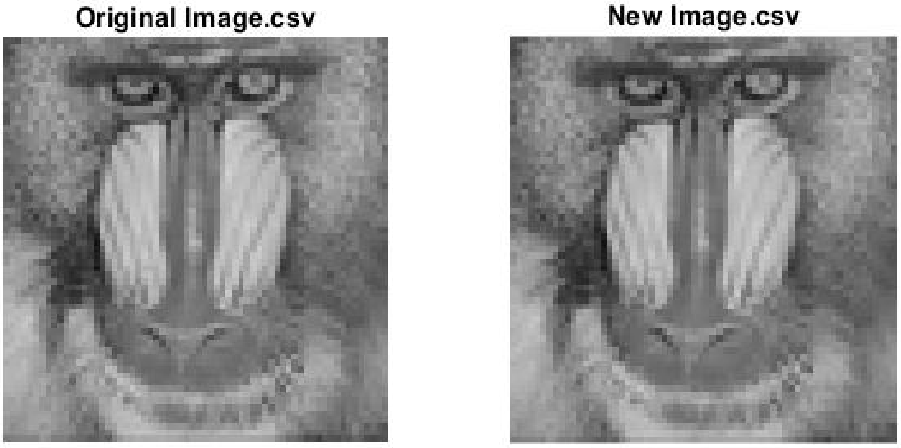
k=6


Original Image.csv
New Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1

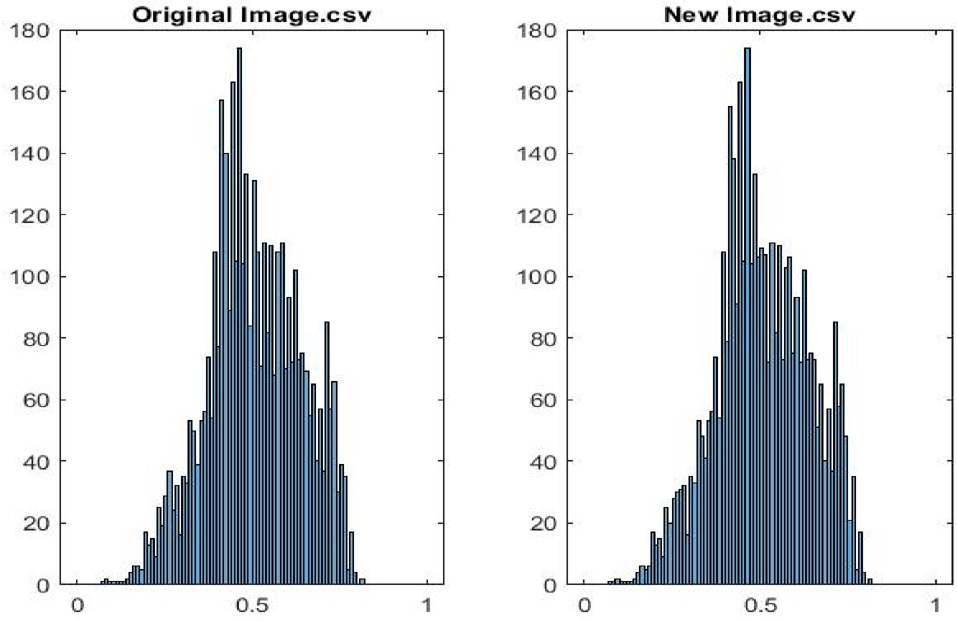

Original Image.csv
New Image.csv

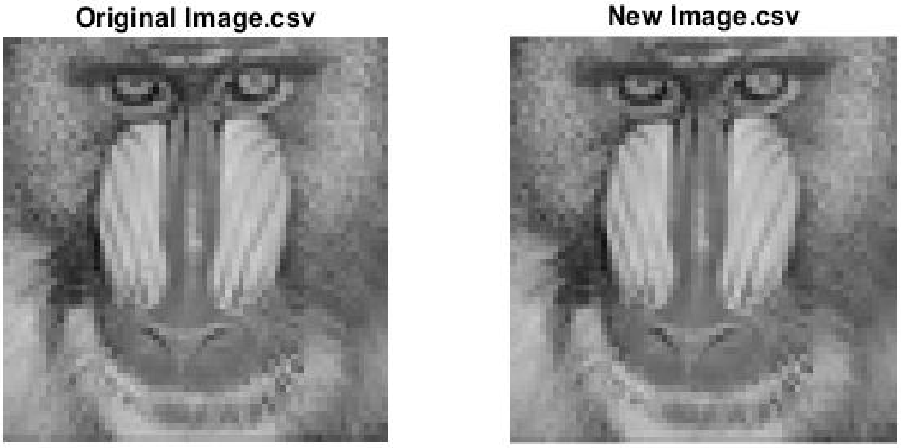
k=7


Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1


Original Image.csv
New Image.csv

k=8


Original Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1
New Image.csv
180
160
140
120
100
80
60
40
20
0
0
0.5
1


Original Image.csv
New Image.csv

(b)

1. (**Bonus**) $\frac{1}{x-y} + \frac{1}{(x+y)^2} \leq z$, $x > |y|$

   **Answer:**

   **Proof of Convex function**:

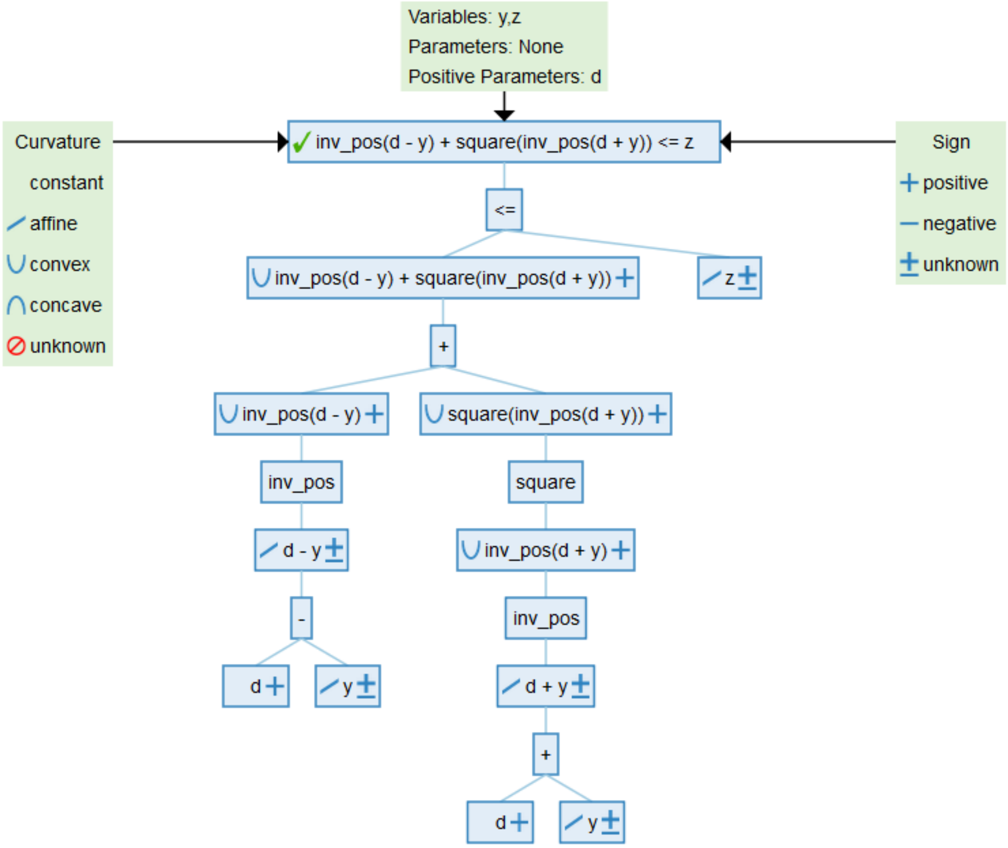
   If we calculate hassian of the of the given equation, we get,

   $$\begin{pmatrix} 2/(\text{x-y})^3 + 6/(x-y)^4 & \text{-2/(x-y)}^3 - 6/(x-y)^4 \\ \text{-2/(x-y)}^3 - 6/(x-y)^4 & 2/(\text{x-y})^3 + 6/(x-y)^4 \end{pmatrix}$$

   Now, this matrix is symmetric ans as $x > |y|$ , it will always positive semi-definite. So, this inequality is convex.

   The DCP Expression used : $inv\_pos(d - y) + square(inv\_pos(d + y)) <= z$

   **Result:**



2. (**Bonus**) $x^2 + \frac{2}{\log^2 y} \leq 5z^{\frac{1}{4}}$, $y > 1$, $z \geq 0$

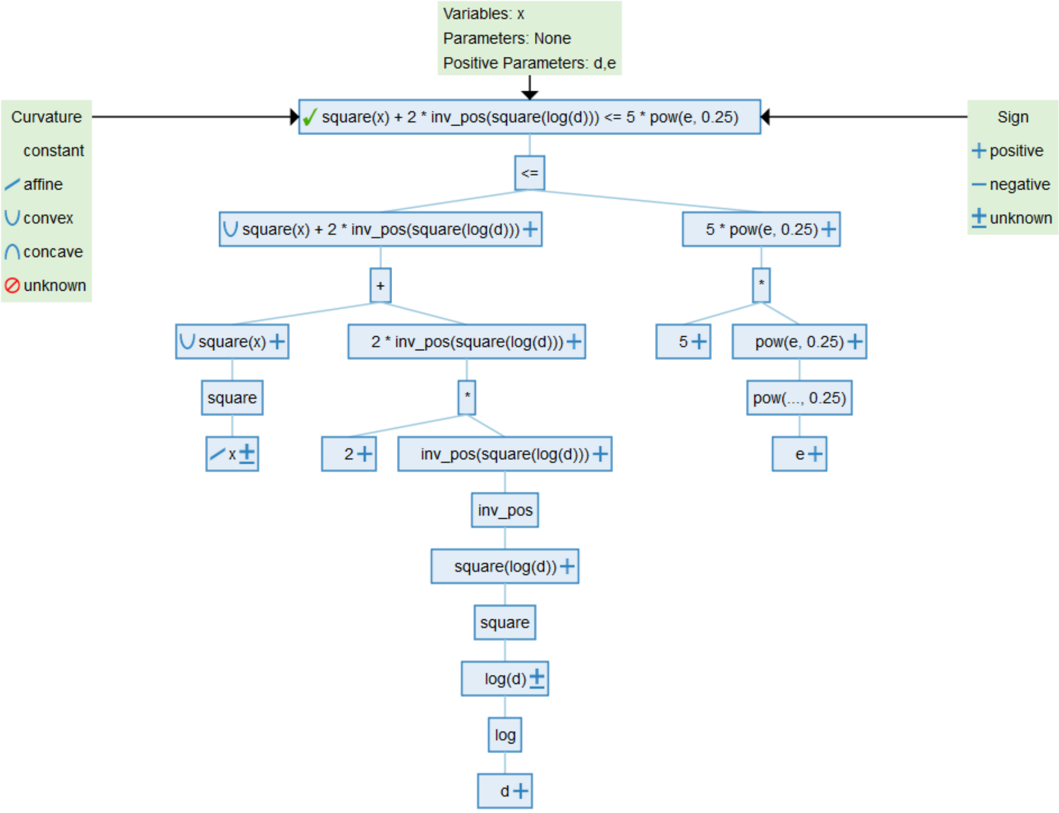
   **Answer:**

   **Proof of Convex function**:

   If we calculate hassian of the of the given equation, we get,

   $$\begin{pmatrix} 2 & 0 \\ 0 & 12/\text{y}^2 log^4 y + 4/y^2 log^3 y \end{pmatrix}$$

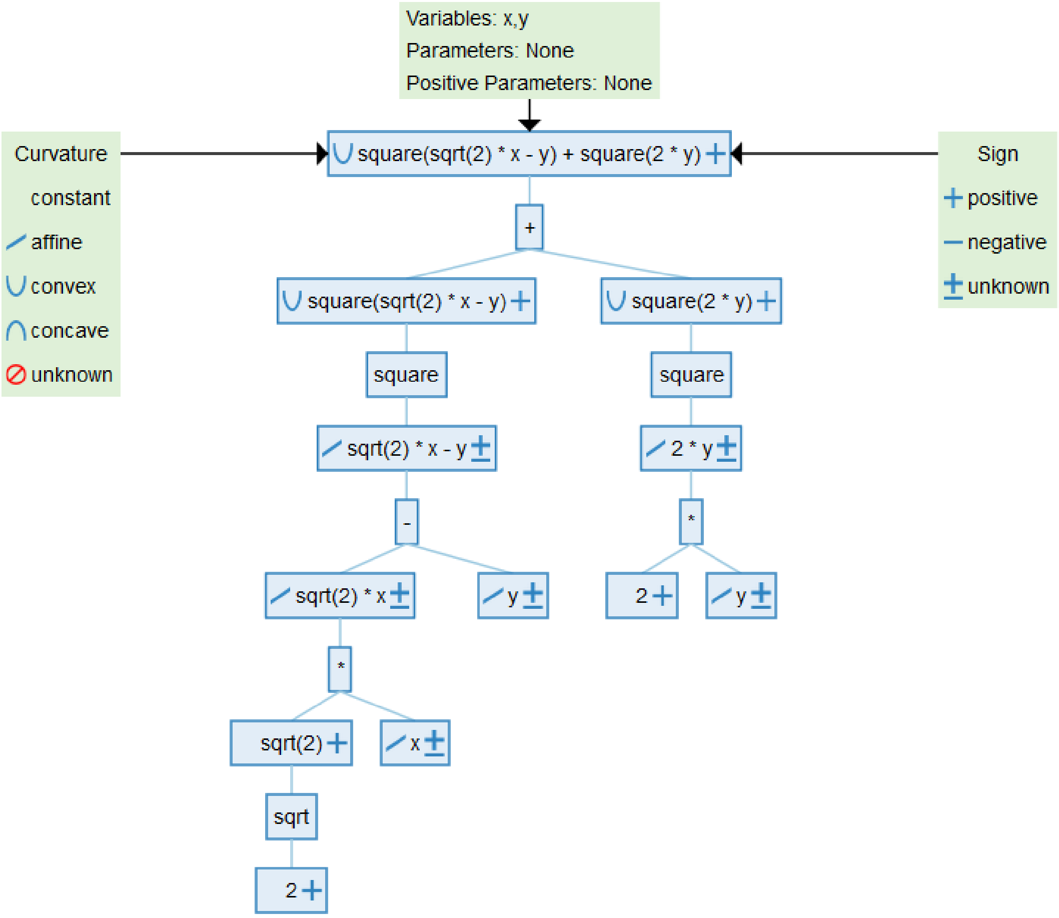
Now, as $y > 1$ , it will always positive semi-definite. So, this inequality is convex.
The DCP Expression used : $square(x) + 2 * inv\_pos(square(log(d))) <= 5 * pow(e, 0.25)$
**Result:**



3. (**Bonus**) $2x^2 - 2xy + 5y^2 = 0$
**Answer:**
**Proof of Convex function**:
If we calculate hassian of the of the given equation, we get,

$$\begin{pmatrix} 4 & -2 \\ -2 & 10 \end{pmatrix}$$

Since, this is Symmetric and a symmetric matrix is positive semidefinite iff the subdeterminants are 0. So the function is convex.
Now for DCP expression we need to modify the function:
$2x^2 - 2xy + 5y^2 = 0$
$x^2 - 2xy + y^2 + x^2 + 4y^2 = 0$
$(x - y)^2 + x^2 + (2y)^2 = 0$
The DCP Expression used : $square(sqrt(2) * x - y) + square(2 * y)$
**Result:**



## 2 Convex sets

Closed and convex sets.

i. Show that If $S \subseteq \mathbb{R}^n$ is convex, and $A \in \mathbb{R}^{m \times n}$, then $A(S) = \{Ax : x \in S\}$, called the image of $S$ under $A$, is convex.

   **Answer:** Given, $S$ is a convex set. Now let's two elements $x$, $y$ from $S$. As the definition of convex set, we know, all elements joining the line segment of $x$ and $y$ is also in the set. It is: $tx + (1-t)y \in S$ when $0 \leq t \leq 1$.

   As $A(S)$ is a image of $S$, we can say every element of $S$ will be in $A(S)$, e.g.
   $A(tx + (1-t)y) \in A(S)$
   or, $tA(x) + (1-t)A(y) \in A(S)$

In other words, line segment joining $A(x)$ and $A(y)$ is also in $A(S$. So, we can say that $A(S) = \{Ax : x \in S\}$ is convex. (Showed)

ii. Show that if $S \subseteq \mathbb{R}^m$ is convex, and $A \in \mathbb{R}^{m\times n}$, then $A^{-1}(S) = \{x : Ax \in S\}$, called the preimage of $S$ under $A$, is convex.

**Answer:** Here given, $A^{-1}(S) = \{x : Ax \in S\}$.
So, for two point $A(x), A(y) \in S$ we will show $x, y \in A^{-1}(S)$.
Now, as $S$ is convex, we can write
$tA(x) + (1-t)A(y) \in S$
or, $A(tx + (1-t)y) \in S$
or, $tx + (1-t)y \in A^{-1}(S)$

So, we can write $A^{-1}(S) = \{x : Ax \in S\}$ the preimage of $S$ under $A$, is convex.(Showed)

iii. Prove that the *log barrier function* $f : \mathbb{R}^n_{++} \to \mathbb{R}$, defined as

$$f(x) = -\sum_{i=1}^{n} \log(x_i),$$

is strictly convex.

**Answer:** To prove the *log barrier function* strictly convex, we need to calculate it's second derivative. If it's second derivative is greater than 0 , than it is strictly convex.
Given function,

$$f(x) = -\sum_{i=1}^{n} \log(x_i),$$

First order derivative is,

$$f'(x) = -\sum_{i=1}^{n}(\frac{1}{x_i}),$$

Second order derivative is,

$$f''(x) = \sum_{i=1}^{n}(\frac{1}{x_i^2}),$$

Which is clearly positive. So we can say the log barrier function is strictly convex.(Proved)

iv. Let $f$ be twice differentiable, with dom$(f)$ convex. Prove that $f$ is convex if and only if

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq 0,$$

for all $x, y$. This property is called *monotonicity* of the gradient $\nabla f$.

**Answer:** Let's differentiate $f$ for two variable x and y,
$f(y) \geq f(x) + \nabla f(x)^T(y-x)$........... (1)
$f(x) \geq f(y) + \nabla f(y)^T(x-y)$........... (2)

Adding equation 1 and 2 we get,
$f(y) + f(x) \geq f(x) + f(y) + \nabla f(x)^T(y-x) + \nabla f(y)^T(x-y)$
or,$f(y) + f(x) \geq f(x) + f(y) - \nabla f(x)^T(x-y) + \nabla f(y)^T(x-y)$
or, $f(y) + f(x) \geq f(x) + f(y) + (\nabla f(y) - \nabla f(x))^T(x-y)$
or, $0 \geq (\nabla f(y) - \nabla f(x))^T(x-y)$
or, $0 \leq (\nabla f(x) - \nabla f(y))^T(x-y)$
or, $(\nabla f(x) - \nabla f(y))^T(x-y) \geq 0$ (Proved)

## 3 Lipschitz gradients and strong convexity

Let $f$ be convex and twice continuously differentiable.

(Part a) Given, $\nabla f$ is Lipschitz with constant $L$, prove i, ii, iii statements.

i. $(\nabla f(x) - \nabla f(y))^T(x-y) \leq L\|x-y\|_2^2$ for all $x, y$;

**Answer:**

We know, A differential function $f$ has an Lipschitz continuous gradient on any set $dom(f)$ if there is a constant $L > 0$
such that:
$||\nabla f(x) - \nabla f(y)|| \leq L||x-y||\ \forall x, y \in dom(f)$
or, $(\nabla f(x) - \nabla f(y))^T \leq L||x-y||^T$
or, $(\nabla f(x) - \nabla f(y))^T(x-y) \leq L(x-y)^T(x-y)$
or, $(\nabla f(x)\nabla f(y))^T(x-y) \leq L||x-y||_2^2$ (Proved)

ii. $\nabla^2 f(x) \preceq LI$ for all $x$;
**Answer:** Again, we know,
$||\nabla f(x) - \nabla f(y)|| \leq L||x-y||$
or,$\frac{||\nabla f(x) - \nabla f(y)||}{||x-y||_2^2} \leq \frac{L||x-y||}{||x-y||_2^2}$
or, $\nabla^2 f(x) \preceq L$
or, $\nabla^2 f(x) \preceq LI$ where $I$ is identity matrix.(Proved)

iii. $f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}\|y-x\|_2^2$ for all $x, y$.

**Answer:**
We know another form of Lipschitz continuous gradient condition is:
$f(y) \leq f(x) + \nabla f(x)^T(y-x) + \frac{L}{2}||y-x||_2^2\ \forall x, y \in dom(f)$
Now, if we consider monotonicity of the gradient for any,

$g(t) = f(x + t(y - x))$.

We can write,
$g'(t) - g'(0) = (\nabla f(x + t(y - x)) - \nabla f(x))^T(y - x) \leq tL||y - x||_2^2$
Now if $g(t)$ is defined for $0 \leq t \leq 0$ to support convexity, then integrating from $t = 0$ to $t = 1$ we get
$g(1) = g(0) + \int_0^1 g'(t)dt$
or, $g(1) \leq g(0) + g'(0) + \frac{L}{2}||y - x||_2^2$
or we can write,
$f(y) \leq f(x) + \nabla f(x)^T(y - x) + \frac{L}{2}||y - x||_2^2$(Proved)

Let $f$ be convex and twice continuously differentiable.
(Part b) Given, $f$ is strongly convex with constant $m$, prove i, ii, iii statements.

i. $(\nabla f(x) - \nabla f(y))^T(x - y) \geq m\|x - y\|_2^2$ for all $x, y$;
**Answer:**
From the rule of strong convexity we know that, a function should be as convex as a quadratic function. Or we can write, for any constant $m > 0$
$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{m}{2}||y - x||_2^2\ \forall x, y$......(i)
Now, from the rule of convexity for any function $g(x)$ we know:
$g(y) \geq g(x) + \nabla g(x)^T(y - x)$, $\forall x, y$
monotonicity of the gradient we can write
$(\nabla g(x) - \nabla g(y))^T(x - y) \geq 0$, $\forall x, y$
Comparing it with equation (i) we can write,
$(\nabla f(x) - \nabla f(y))^T(x - y) \geq m||x - y||_2^2$,$\forall x, y$
It also supports strong convexity rule. (Proved)

ii. $\nabla^2 f(x) \succeq mI$ for all $x$;
**Answer:**
From the previous part we can write,
$||\nabla f(x) - \nabla f(y)|| \geq m||x - y||$
or, $\frac{||\nabla f(x) - \nabla f(y)||}{||x-y||_2^2} \geq \frac{m||x-y||}{||x-y||_2^2}$
or, $\nabla^2 f(x) \geq m$
or, $\nabla^2 f(x) \succeq mI$ [Where $I$ is the Identity matrix](Proved)

iii. $f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{m}{2}\|y - x\|_2^2$ for all $x, y$.

**Answer**: We know, from the rule of convex function,
$f(tx + (1 - t)y) \leq tf(x) + (1 - t)f(y)$ where $0 \leq t \leq 1$
We have already proved that, for strong convex function, for any constant $m > 0$
$(\nabla f(x) - \nabla f(y))^T(x - y) \geq m||x - y||^2$
or, $f(y) \geq f(x) + (x)^T(y - x) + \frac{m}{2}||x - y||^2$(Proved)

# 4 Subgradients and Proximal Operators

1. Determine the subdifferential of $||x||_1$ and $||x||_2$.

   **Answer: For l1 Norm:** In case of $f(x) = ||x||_1$, it is non differential when $x = 0$. We can calculate gradient descent when $x \neq 0$. So the rule will be finding gradient for differential portions and letter continuing them for non-differentiable parts.
   Now let express the function $f(x)$ as the maximum of $2^n$ linear functions:
   $||x||_1 = max\{s^T x | s_i \in \{-1, 1\}\}$,
   Here we will find an $s \in \{1, +1\}^n$ such that $s^T x = ||x||_1$.
   So, for$x_i > 0$, $s_i = +1$
   For, $x_i < 0$. $s_i = 1$
   and for, $x_i = 0$, we will consider both function, $s_i = +1$, and $s_i = 1$.
   So, we can write,

$$g_i = max \begin{cases} +1 & ; x_i > 0 \\ -1 & ; x_i < 0 \\ -1/+1 & ; x_i = 0 \end{cases}$$

   So our subgradient can be:
   $\partial f(x) = g : ||g||_\infty \leq 1, g^T x = ||x||_1$

   **For l2 Norm:**
   We know for the two norm,
   $f(x) = ||x||_2 = \sqrt{x_1^2 + x_2^2 + .... + x_n^2}$
   Similar to previous case, the function is non-differentiable for $x = 0$ and differentiable when $x \neq 0$.
   So we can write,

$$\partial f(x) = \begin{cases} \frac{x}{||x||_2} & ; x \neq 0 \\ g : ||g||_2 \leq 1 & ; x = 0 \end{cases}$$

2. The proximal operator for function $h : \mathbb{R}^n \mapsto \mathbb{R}$ and $t > 0$ is defined as:

$$\text{prox}_{h,t}(x) = \underset{z}{\text{argmin}} \frac{1}{2}||z - x||_2^2 + th(z)$$

   Compute the proximal operators $\text{prox}_{h,t}(x)$ for the following functions.

   (i) $h(z) = \frac{1}{2}z^T Az + b^T z + c$, where $A \in \mathbb{S}_+^n$.
   **Answer:**$A \in S_+^n$, So $A \succeq 0$

   $Prox_{t,h}(z) = (I + tA)^{-1}(z - tb)$

   (ii) $h(z) = ||z||_2$.

**Answer:**

$$Prox_{t,h}(z) = max\begin{cases} (1 - \frac{t}{||z||_2})z & ; ||z||_2 \geq t \\ 0 & ; Otherwise \end{cases}$$

(**Bonus**)(iii) $h(z) = ||z||_0$, where $||z||_0$ is defined as $||z||_0 = |\{z_i : z_i \neq 0, i = 1, \ldots, n\}|$.

**Answer**:
Here given function,$h(z) = ||z||_0$

$$Prox_{t,h}(z_i)\forall i \in [1-n] = max\begin{cases} \{0\} & ; z_i < \sqrt{2} \\ \{z_i\} & ; z_i > \sqrt{2} \\ \{0, z_i\} & ; z_i = \sqrt{2} \end{cases}$$

and final proxima will be,
$Prox_{t,h}(z) = \Pi_{i=1}^{n} Prox_{t,h}(z_i)\forall i \in [1-n]$

# 5 Convergence of Gradient Descent

In this problem, you will show the sublinear convergence for gradient descent, which was presented in class.

To be clear, we assume that the objective $g(x)$ satisfy:

(A1) $g$ is convex, differentiable, and $\text{dom}(g) = \mathbb{R}^n$.

(A2) $\nabla g$ is Lipschitz, with constant $L > 0$.

(a) (Bonus) We will prove that the gradient descent converges sublinearly in this case. As a reminder, the iterates of gradient descent is computed by

$$x^+ = x - t\nabla g(x), \tag{1}$$

where $x^+$ is the iterate succeeding $x$. Henceforth, we will set $t = 1/L$ for simplicity.

(i) (Bonus) Show that

$$g(x^+) - g(x) \leq -\frac{1}{2L}\|\nabla g(x)\|^2.$$

That is, the objective value is monotonically decreasing in each update. This is why gradient descent is called a "descent method."

**Answer:**
Considering, g is convex, differentiable, and $dom(g) = R^n$. We will have
$\nabla^2 g(x) \preceq LI$ for a constant $L > 0$
and there will be,
$\nabla^2 g(x) - LI$ is a negative semidefinite matrix.

Using quadratic expansion on $g(x)$:
$g(y) \leq g(x) + \nabla(x)^T(y-x) + \frac{1}{2}\nabla^2 g(x)||y-x||^2$
or, $g(y) =\leq g(x) + \nabla g(x)^T(y-x) + \frac{1}{2}L||y-x||^2$
Now replacing $y = x^+$, and fixed step size $t = \frac{1}{L}$ we get.
$g(x^+) \preceq g(x) + \nabla g(x)^T(x^+ - x) + \frac{1}{2}L||x^+ - x||^2$
Now replacing, $x^+ = x - t\nabla g(x)$ we get
$g(x^+) \preceq g(x) + \nabla g(x)^T(x - t\nabla g(x) - x) + \frac{1}{2}L||x - t\nabla g(x) - x||^2$
or,$g(x^+) \preceq g(x) - \nabla g(x)^T t\nabla g(x) + \frac{1}{2}L||t\nabla g(x)||^2$
or, $g(x^+) \preceq g(x) - t||\nabla g(x)||^2 + \frac{1}{2}Lt^2||\nabla g(x)||^2$
or,$g(x^+) \preceq g(x) - t(1 - \frac{1}{2}Lt)||\nabla g(x)||^2$
Here replacing $t = \frac{1}{L}$ we get
$t(1 - \frac{1}{2}Lt) = \frac{1}{2L}$
So,
$g(x^+) \leq g(x) - \frac{1}{2L}||\nabla g(x)||^2$
$g(x^+) - g(x) \leq \frac{1}{2L}||\nabla g(x)||^2$(Showed)

(ii) (Bonus) Using convexity of $g$, show the following helpful inequality:

$$g(x^+) - g(z) \leq \nabla g(x)^T(x-z) - \frac{1}{2L}\|\nabla g(x)\|^2, \quad \forall z \in \mathbb{R}^n.$$

**Answer:** From part (i), we have -
$g(x^+) - g(x) \leq -\frac{1}{2L}||\nabla g(x)||^2$
or, $g(x^+) - g(x) + \frac{1}{2L}||\nabla g(x)||^2 \leq 0$
Now, again adding to both side $g(z) = g(x) + \nabla g(x)^T(z-x)$, where, $\forall z \in R^n$ we get
or, $(g(x^+) - g(x) + \frac{1}{2L}||\nabla g(x)||^2) + (g(x) + \nabla g(x)^T(z-x)) \leq 0 + g(z)$
$g(x^+) + \frac{1}{2L}||\nabla g(x)||^2 + \nabla g(x)^T(z-x)) \leq g(z)$
or, $g(x^+) - g(z) \leq -\frac{1}{2L}||\nabla g(x)||2 - \nabla g(x)^T(z-x)$
or,$g(x^+) - g(z) \leq \nabla g(x)^T(x-z) - \frac{1}{2L}||\nabla g(x)||^2$(Showed)

(iii) (Bonus) Show that

$$g(x^+) - g(x^\star) \leq \frac{L}{2}\left(\|x - x^\star\|^2 - \|x^+ - x^\star\|^2\right),$$

where $x^\star$ is the minimizer of $g$, assuming $g(x^\star)$ is finite.
**Answer:**
The positive nature of $t^2||\nabla g(x)||_2^2$ as the gradint will decrease it will reach to the optimum, where, $g(x) = g(x^*)$. If we write $g(x^+)$, in terms of $g(x^*)$, we get
$g(x^*) \leq g(x) + \nabla g(x)^T(x^* - x)$
$g(x) \geq g(x^*) + \nabla g(x)^T(x - x^*)$

We already know,
$g(x^+) \leq g(x) - \frac{1}{2L}||\nabla g(x)||^2$
As $g(x)$ is at least bigger than $\leq g(x) - \frac{1}{2L}||\nabla g(x)||^2$, we can replace $g(x)$,
$g(x^+) \leq g(x^*) + \nabla g(x)^T(x - x^*) - \frac{t}{2}||\nabla g(x)||^2$
or, $g(x^+) - g(x^*) \leq \nabla g(x)^T(x - x^*) - \frac{t}{2}||\nabla g(x)||^2$

or, $g(x^+) - g(x^*) \leq \frac{1}{2t}(2t\nabla g(x)^T(x - x^*) - t^2||\nabla g(x)||^2$
or, $g(x^+) - g(x^*) \leq \frac{1}{2t}(2t\nabla g(x)^T(x - x^*) - t^2||\nabla g(x)||^2 + ||x - x^*||^2 - ||x - x^*||^2)$
or, $g(x^+) - g(x^*) \leq \frac{1}{2t}(||x - x^*||^2 - (||x - x^*||^2 + t^2||\nabla g(x)||_2^2 - 2t\nabla g(x)^T(x - x^*)))$
or, $g(x^+) - g(x^*) \leq \frac{1}{2t}(||x - x^*||^2 - ||(x - x^*) - t\nabla g(x)^T||^2)$
or, $g(x^+) - g(x^*) \leq \frac{1}{2t}(||x - x^*||^2 - ||x - t\nabla g(x)^T - x^*||^2)$
We know $x^+ = x - t\nabla g(x)$,
So,
$g(x^+) - g(x^*) \leq \frac{1}{2t}(||x - x^*||^2 - ||x^+ - x^*||^2)$
or, $g(x^+) - g(x^*) \leq \frac{1}{2\frac{1}{L}}(||x - x^*||^2 - ||x^+ - x^*||^2)$
or,$g(x^+) - g(x^*) \leq \frac{L}{2}(||x - x^*||^2 - ||x^+ - x^*||^2)$(Showed)

(iv) (Bonus) Now, aggregating the last inequality over all steps $i = 0, \ldots, k$, show that the accuracy of gradient descent at iteration $k$ is $O(1/k)$, i.e.,

$$g(x^{(k)}) - g(x^\star) \leq \frac{L}{2k}\|x^{(0)} - x^\star\|^2.$$

Put differently, for an $\epsilon$-level accuracy, you need to run at most $O(1/\epsilon)$ iterations.

**Answer:** If we consider 5(iii) over all $i = 0, ..., k$, we get -
$g(x^k) - g(x^*) \leq \frac{1}{k}\sum_{i=1}^k g(x^i) - g(x^*)$ [since g decreases on every iteration]
or, $g(x^k) - g(x^*) \leq \frac{1}{k} \times \sum_{i=1}^k \frac{1}{2t}(||x^{i-1} - x^*||^2 - ||x^i - x^*||^2)$
or, $g(x^k) - g(x^*) \leq \frac{1}{k} \times \frac{1}{2t}(||x^0 - x^*||^2 - ||x^k - x^*||^2)$
or, $g(x^k) - g(x^*) \leq \frac{1}{k} \times \frac{1}{2t}(||x^0 - x^*||^2)$
or, $g(x^k) - g(x^*) \leq \frac{L}{2k}||x^0 - x^*||^2$
So, the accuracy of gradient descent at iteration $k$ is $O(1/k)(Showed)$

# 6 Proximal Gradient Descent

1) We first consider the ridge regression problem, where $h(\beta) = \frac{\lambda}{2}\|\beta\|_2^2$:

$$\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{2N}\|X\beta - y\|^2 + \frac{\lambda}{2}\|\beta\|_2^2 \tag{2}$$

where $N$ is the number of samples. Note: in your implementation for this problem, if you added a ones vector to $X$ ($X = \begin{bmatrix}\mathbf{1} & X_{(1)} & X_{(2)} & \ldots & X_{(J)}\end{bmatrix}$), **you should not include the bias term $\beta_0$ associated with the ones vector in the penalty.**

(a) (**Bonus**) Use **CVX** to solve the ridge regression problem (2). Initialize $\beta$ with random normal values. Fit the model parameters on the training data (`X_train.csv`, `Y_train.csv`) and evaluate the objective function .Set $\lambda = 1$. Compare the solution to the one you get from part 2).

**Answer**:
The code is implemented on code file Question_6_1.m. Here the result we obtain is 57.041. Which is higher than the value obtained in part 2, that is 54.8943.

2) Next, we consider the least squares group LASSO problem, where $h(\beta) = \lambda \sum_j w_j \|\beta_{(j)}\|_2$:

$$\min_{\beta \in \mathbb{R}^{p+1}} \frac{1}{2N}\|X\beta - y\|^2 + \lambda \sum_j w_j \|\beta_{(j)}\|_2 \tag{3}$$

A common choice for weights on groups $w_j$ is $\sqrt{p_j}$, where $p_j$ is number of predictors that belong to the $j$th group, to adjust for the group sizes. Additionally, the same as in 1), **do not include $\beta_0$ in the penalty.**

We will solve the problem using proximal gradient descent algorithm (over the whole dataset).

(a) Derive the proximal operator $\text{prox}_{h,t}(x)$ for the non-smooth component $h(\beta) = \lambda \sum_{j=1}^{J} w_j \|\beta_{(j)}\|_2$.

**Answer**:
Here,
$\text{prox}_{h,t}(\beta) = min\{\beta - t\nabla g(\beta)\}$
Here, $\nabla g(\beta) = \lambda \sum_{j=1}^{J} w_j \frac{\beta_{(j)}}{\|\beta_{(j)}\|_2}$ [Considering proximal for euclidean norm] So, our overall equation will be,
$\text{prox}_{h,t}(\beta) = \beta - t\lambda \sum_{j=1}^{J} w_j \frac{\beta_{(j)}}{\|\beta_{(j)}\|_2}$

(b) Derive the proximal gradient update for the objective.

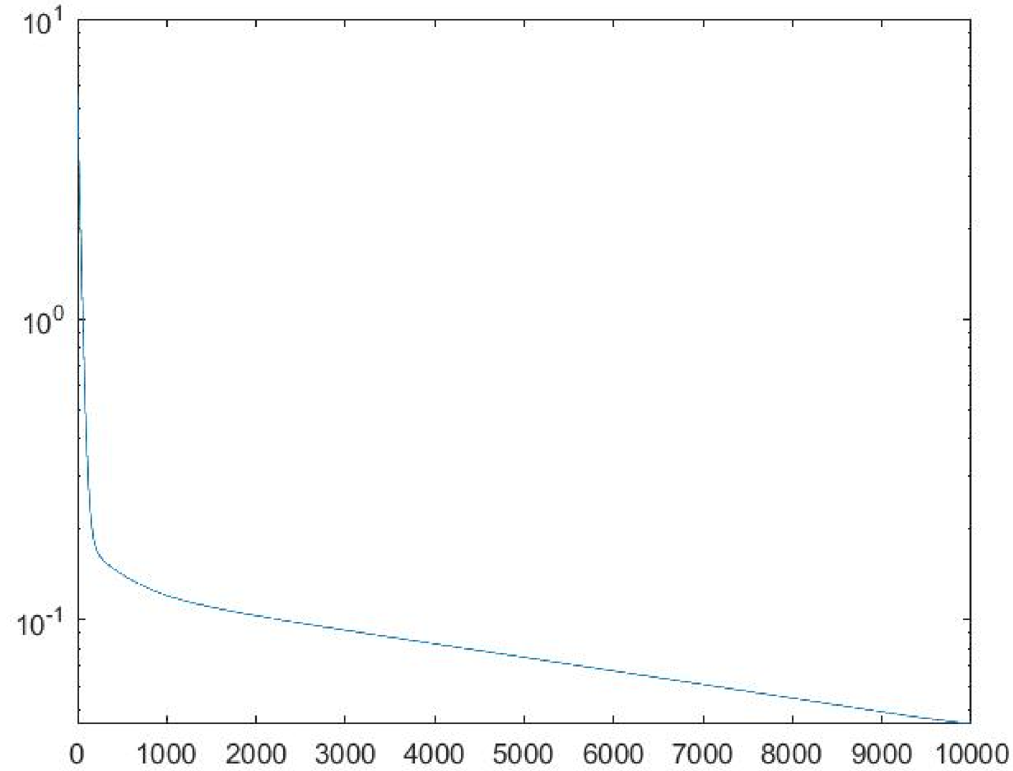
**Answer**:
Here, proximal update $\beta+$ will be,
$\text{prox}_{h,t}(\beta) = \beta - t\nabla g(\beta)$
$\text{prox}_{h,t}(\beta) = \beta - \frac{t}{2N} X^T(X\beta - y))$

(c) Implement proximal gradient descent to solve the least squares group lasso problem on the `Parkinsons` dataset. Set $\lambda = 0.02$. Use a fixed step-size $t = 0.005$ and run for 10000 steps.

**Answer**: The implemented code is available on code file $Question_6\_2\_cd.m$.

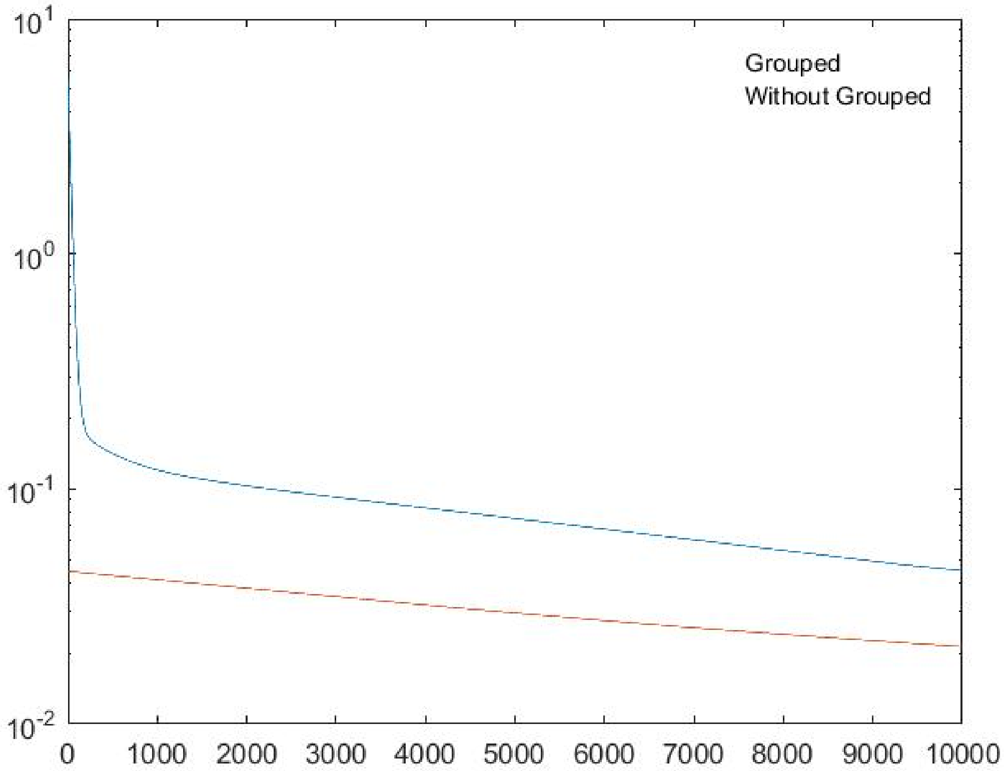
(d) Plot $f^k - f^\star$ versus $k$ for the first 10000 iterations ($k = 1, \ldots, 10000$) on a semi-log scale (i.e. where the y-axis is in log scale) for the training data, where $f^k$ denotes the objective value averaged over all samples at step $k$, and the optimal objective value is $f^\star = 49.9649$. Print the components of the solutions numerically. What are the selected groups?

**Answer**
Here the 18 features are grouped in 8 groups, first 2, as single, next is 6, next is 5, next is 2 and final three as single. Another one represents the beta 0.
Here, average of all components we get is 54.8943
The implemented code is available on code file $Question_6\_2\_cd.m$.



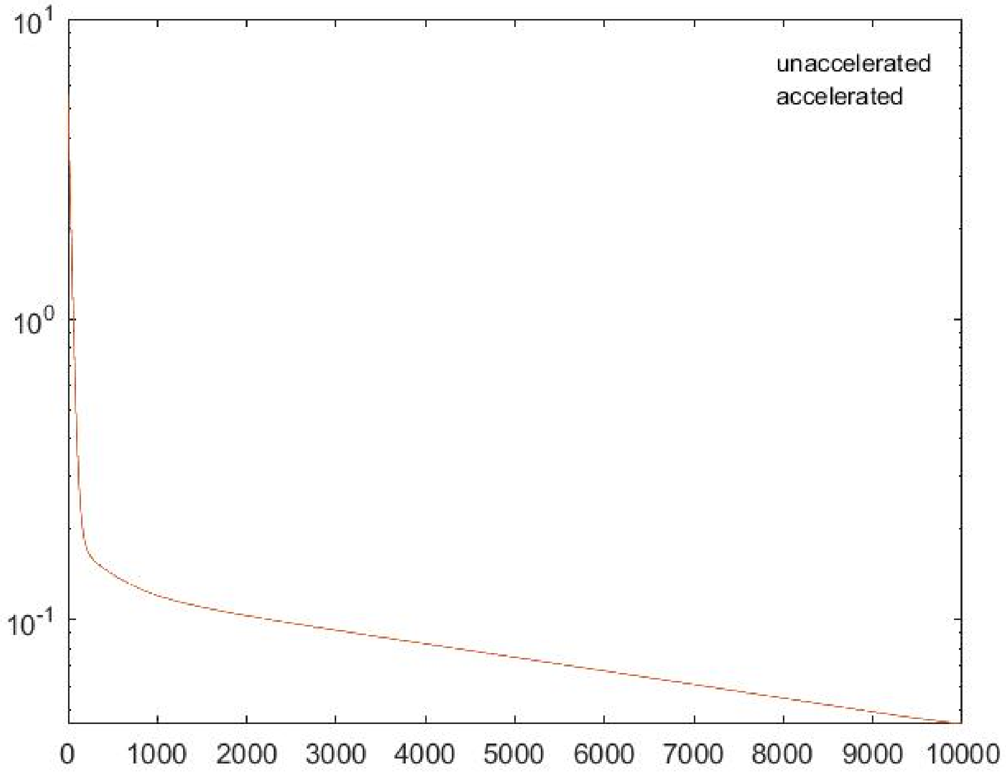
(e) Now implement the LASSO (hint: you shouldn't have to do any additional coding), with fixed step-size $t = 0.005$ and $\lambda = 0.02$. Compare the LASSO solution with your group lasso solutions.

**Answer**:
The implemented code is available on code file $Question_6\_2\_e.m$
Here we can see that the non grouped curve kind affine and go fur below the grouped curve.



(f) (**Bonus**) Implement accelerated proximal gradient descent with fixed step-size under the same setting in part (c). Hint: be sure to exclude the bias term $\beta_0$ from the proximal update, just use a regular accelerated gradient update. Plot $f^k - f^\star$ versus $k$ for both methods (unaccelerated and accelerated proximal gradient) for $k = 1, \ldots, 10000$ on a semi-log scale and compare the selected groups. What do you find?

**Answer**:
The implemented code is available on code file $Question_6\_2\_f.m$
Here we can see that both curve following the exactly the same curve


10^1
10^0
10^-1
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000
unaccelerated
accelerated