```m
clc; clear all; close all;
%%
dir = 'videos/';
v = VideoReader([dir '1.mp4']); % change to see video 1, 2, 3

dt = 1/30; % 30 fps from settings

% store the video frame by frame
count = 0;
while hasFrame(v)
    count = count + 1;
    img = readFrame(v);
    img = double(rgb2gray(img));
    images{count} = img;
end

[~, T] = size(images);
t = 0:dt:T;
%% Vectorize
[col, row] = size(images{1});
for i = 1:size(images,2)
    data(:, i) = reshape(images{i}, col * row, 1);
end

%% DMD Algorithm
X1 = data(:, 1:end-1);
X2 = data(:, 2:end);

% SVD
[U, S, V] = svd(double(X1), 'econ');

%% Rank Truncuation if needed
% r = 100;
% U = U(:, 1:r);
% S = S(1:r, 1:r);
% V = V(:, 1:r);
%% Get the model
Atide = U'*X2*V/S;
[W, D] = eig(Atide);
Phi = X2 * V / S * W; % DMD modes
omega = log(diag(D)) / dt;
b = Phi \ X1(:,1);

%% Compute for XLowRank and XSparse
[~,p] = min(abs(omega));

% Construct the low-rank matrix for every time step by using only the mode
% associated with the minimum frequency.
for k = 1:size(data,2)
    XLowRank(:,k) = Phi(:,p)*diag(exp(omega(p).*t(k)))*b(p);
end

XSparse = data - abs(XLowRank);

%% Find negative values
R = XSparse;
R(R>0) = 0;

XLowRank = R + abs(XLowRank);
XSparse = XSparse - R;

%% Reshape Back to 3D
for i = 1:size(XSparse,2)
    newVideo{i} = reshape(XSparse(:, i), col, row);
end
```

```
66
67
68  %% See the new video
69  % figure(1)
70  % for i = 1:size(newVideo,2)
71  %     imshow(newVideo{i});
72  %     pause(0.03)
73  % end
74  %% Comparison
75  figure(2)
76  frame = 80;
77  count = 0;
78  v = VideoReader([dir '1.mp4']);
79  while hasFrame(v)
80      if count < frame
81          count = count + 1;
82          imag = readFrame(v);
83      elseif count == frame
84          imag = readFrame(v);
85          break;
86      end
87  end
88  subplot(1, 2, 1), imshow(imag), title('Original Frame');
89  subplot(1, 2, 2), imshow(real(newVideo{frame})), title('Frame without background');
```