

AdvLaneFind-Copy1

June 19, 2020

0.0.1 Step 1- Camera Calibration

I began by Preparing Objpoints and ImagePoints. The accompanying advances were followed:-

1. Grayscale the picture
2. Discover Chessboard Corners. It returns two qualities ret,corners. ret stores whether the corners were returned or not
3. In the event that the corners were found, annex corners to picture focuses.
4. I have additionally attracted the chessboard corners to envision the corners

With this progression we will have the option to get picture focuses and object focuses which will be required to compute the camera alignment and bending coefficients.

We call the calibrateCamera work which returns us a lot of boundaries, however the ones we are intrigued are the camera network (mtx) and contortion coefficient (dist).

We at that point utilize the contortion coefficient to undistort our picture.

- From Step 1 we will get the Object Points and Image Points

In [35]: #First thing first

```
# Step 1- Camera calibration

#Creating an array for object Points
objp = np.zeros((6*9,3), np.float32)
objp[:, :2] = np.mgrid[0:9, 0:6].T.reshape(-1,2)

opoints=[] #real space 3D points
ipoints=[] #image space 2D points

#list of calibration images
images = glob.glob('camera_cal/calibration*.jpg')

f,axes= plt.subplots(1,2,figsize=(30,30))
```

```

for i,img in enumerate(images):
    originalImage= cv2.imread(img)
    grayImg= cv2.cvtColor(originalImage, cv2.COLOR_BGR2GRAY) #converting to Grayscale b

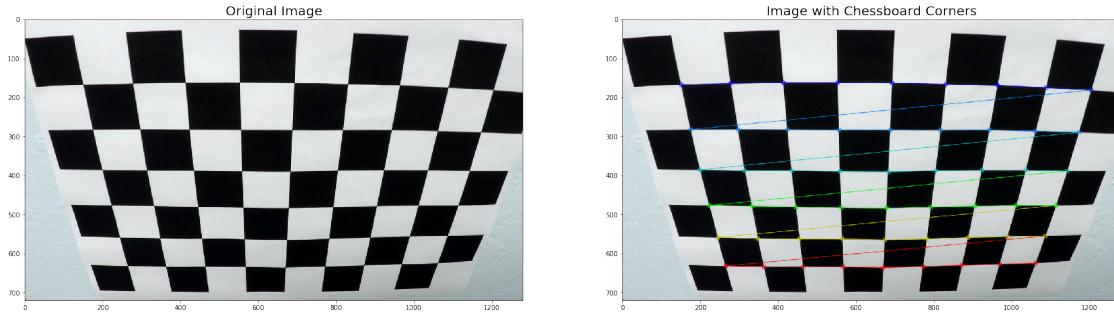
    if(i==1):
        # Plotting the original Image
        axes[0].set_title('Original Image', fontsize=20)
        axes[0].imshow(originalImage)

    # Find the chessboard corners
    ret, corners = cv2.findChessboardCorners(grayImg, (9,6), None)

    if(ret==True):
        opoints.append(objp)
        ipoints.append(corners)

    # Drawing Chessboard Corners
    cv2.drawChessboardCorners(originalImage, (9,6), corners, ret)
    if(i==1):
        axes[1].set_title('Image with Chessboard Corners', fontsize=20)
        axes[1].imshow(originalImage)

```



0.0.2 Step 2- Calculate Undistortion Parameters

- From Step 2 we will get two parameters- dist(the distortion coefficient), mtx(camera matrix)

In [36]: `img = cv2.imread('camera_cal/calibration1.jpg')`
`img_size = (img.shape[1], img.shape[0])`

```

ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(opoints, ipoints, img_size,None,None)

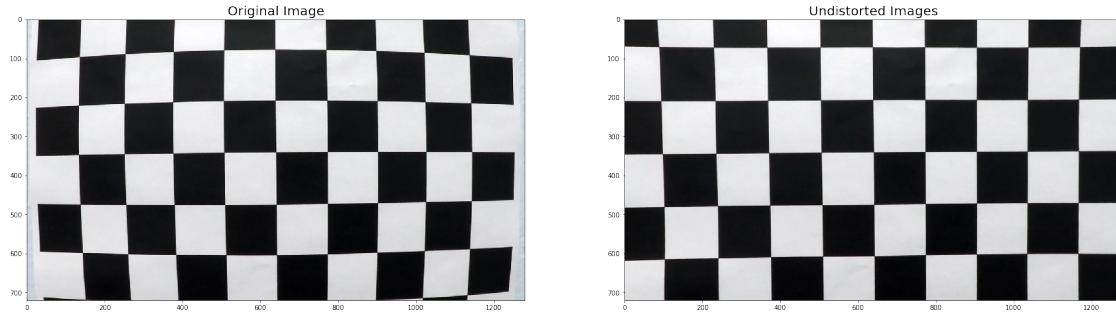
dst = cv2.undistort(img, mtx, dist, None, mtx)

f, axes= plt.subplots(1,2,figsize=(30,30))

```

```
axes[0].imshow(img)
axes[0].set_title("Original Image", fontsize=20)
axes[1].imshow(dst)
axes[1].set_title("Undistorted Images", fontsize=20)
```

Out [36]: Text(0.5,1,'Undistorted Images')



0.0.3 Step 3- Examples of Undistorted Images

In [37]: # Step 3- Defining a function to undistort test Images using parameters derived from pr



0.1 Perspective Transform with Example

In this progression, I initially characterized a Region Of Interest (ROI) for example a Trapezoid with four focuses:

1. Left Bottom Corner characterized as "left"
2. Right Bottom Corner characterized as "right"
3. Left Upper Corner characterized as "apex_left"
4. Right Upper Corner characterized as "apex_right"

Subsequent to characterizing the ROI, the following stage is to twist the picture, to see the picture from elevated point of view.

To do this we have to compute a Matrix with the source and goal focuses. The goal focuses were chosen properly in order to see a decent 10,000 foot point of view. The choice of these focuses depended on hit a preliminary instrument in particular.

When we get the Matrix we will that alongside Image to CV2 warpPerspective capacity to get the last distorted picture.

0.1.1 Step 4 and 5- Region of Interest and Warping

In [38]: # Step 4 and 5- Defining a Region of Interest, Warping an Image from bird's eye view

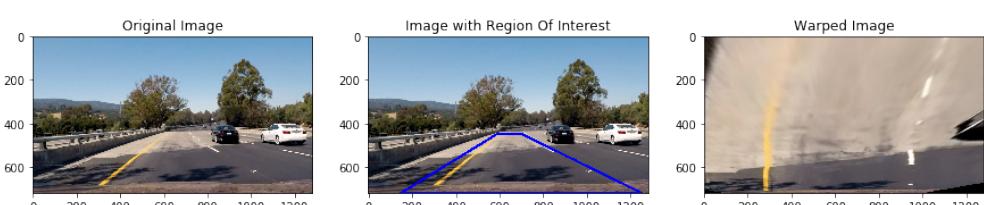
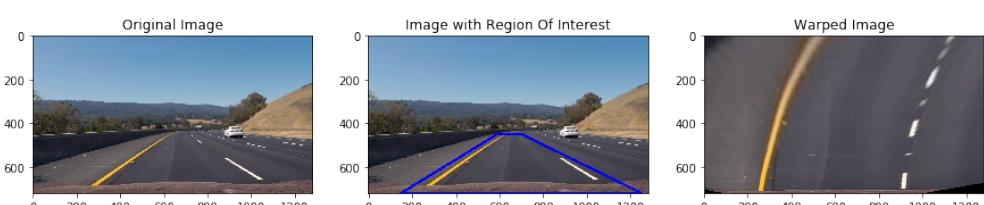
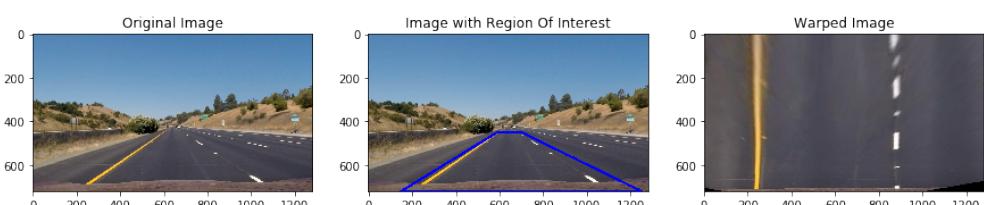
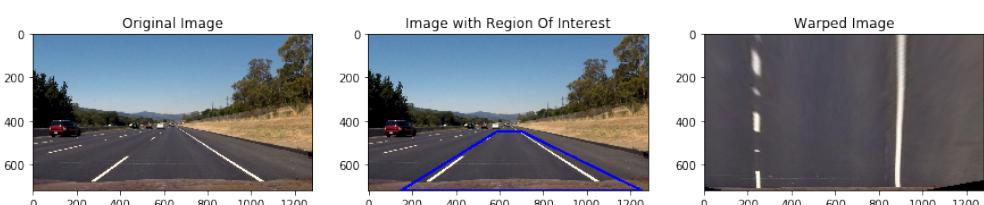
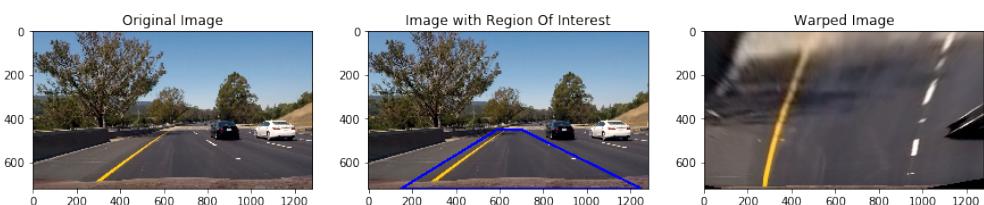
```
left=[150,720] #left bottom most point of trapezium
right=[1250,720] #right bottom most point of trapezium
apex_left=[590,450] # left top most point of trapezium
apex_right=[700,450] # right top most point of trapezium

src=np.float32([left,apex_left,apex_right,right]) # Source Points for Image Warp
dst= np.float32([[200 ,720], [200 ,0], [980 ,0], [980 ,720]]) # Destination Points for Image Warp

def ROI(originalImage):
    return cv2.polylines(originalImage,np.int32(np.array([[left,apex_left,apex_right,right]])),True,(0,255,0),2)

def WarpPerspective(image):
    y=image.shape[0]
    x=image.shape[1]
    M = cv2.getPerspectiveTransform(src, dst)
    return cv2.warpPerspective(image, M, (x,y), flags=cv2.INTER_LINEAR)
```

In [39]: # Testing Region of Interest and Wrap on Test Images



0.1.2 Step 6- Color Space Transformation with example

According to Udacity's proposal, I attempted a different number of colorspace to get a decent twofold picture in all lighting conditions. I attempted the accompanying shading Spaces:-

1. HLS
2. HSV
3. LAB
4. YUV
5. YCrCb

I characterized a typical capacity to remove a specific channel from a colorspace.

Capacity Name-ExtractChannel

Input

- picture - the distorted picture from which we have to remove
- colorspace-the cv2 colorspace. Ex-cv2.COLOR_RGB2HSV
- edge the limit estimation of pixels which should be chosen so as to get the twofold picture. [min_threshold, max_threshold]
- channel-the channel we have to separate from the picture

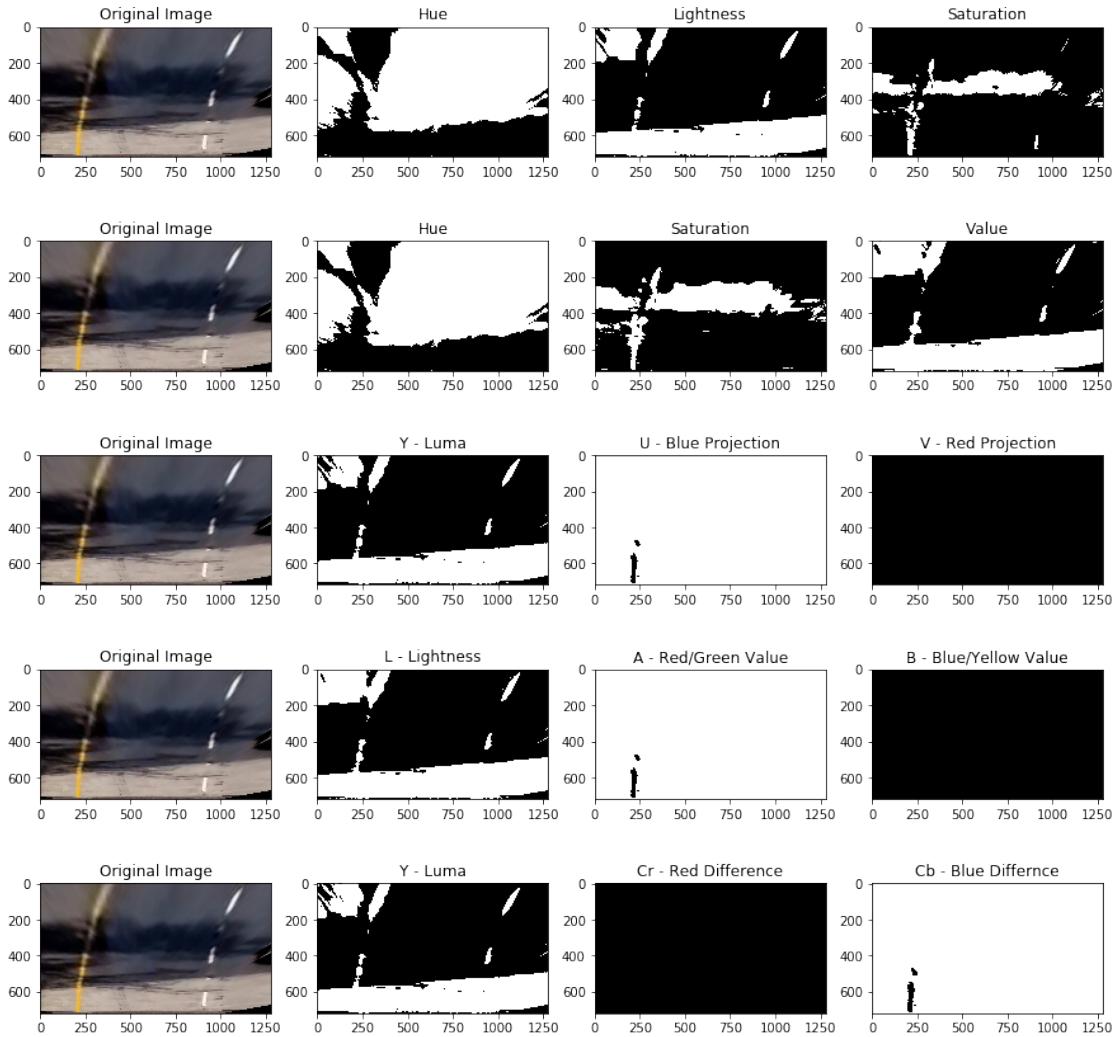
Output

- Paired Image with the necessary channel and edge esteem applied

In [40]: # Step 6- Selecting a Color Space

```
def ExtractChannel(image,colorspace,threshold,channel=0):  
    colorspace = cv2.cvtColor(image, colorspace)  
    extracted_channel = colorspace[:, :, channel]  
    binary = np.zeros_like(extracted_channel)  
    binary[(extracted_channel >= threshold[0]) & (extracted_channel <= threshold[1])] = 1  
    return binary
```

Out [40]: Text(0.5,1,'Cb - Blue Differnce')



0.1.3 Step 7 - Sobel with example

I characterized a typical capacity to apply sobel.

Capacity Name-Sobel

Information

- warpedimage-the first twisted picture
- limit the edge that will be applied to choose the pixel esteem
- sobelType-the bearing where we have to take the slope. values-x-for x slope , y-for y inclination, xy for supreme and dir for bearing
- kernelSize-the size of the portion

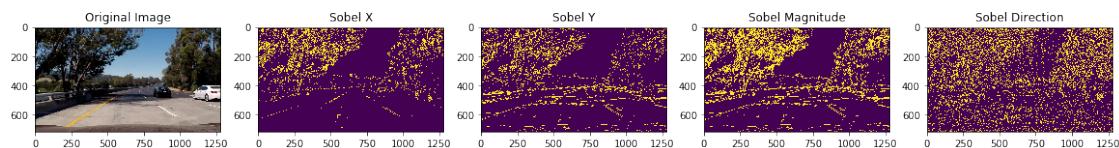
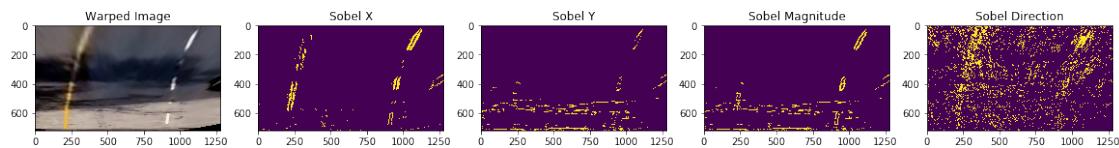
Yield Binary Image with the necessary limits , sobelType and kernelSize

```
In [10]: # Step 7- Applying Sobel to warped image
```

```
def Sobel(warpedimage, threshold, sobelType, kernelSize=3):  
  
    gray = cv2.cvtColor(warpedimage, cv2.COLOR_RGB2GRAY) # Step 1- Convert to GrayScale  
    sobelx = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=kernelSize)  
    sobely = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=kernelSize)  
  
    abs_sobelx = np.absolute(sobelx)  
    abs_sobely = np.absolute(sobely)  
  
    grad= np.sqrt(sobelx**2 + sobely**2)  
  
    arctan= np.arctan2(abs_sobely,abs_sobelx)  
  
    valParam=abs_sobelx  
  
    if(sobelType=='x'):  
        valParam=abs_sobelx  
    elif(sobelType=='y'):  
        valParam= abs_sobely  
    elif(sobelType=='xy'):  
        valParam= grad  
    else:  
        valParam=arctan  
  
  
    img = np.uint8((valParam* 255)/np.max(valParam)) # Creating a normalized sobel image  
    binary_output = np.zeros_like(img)  
    binary_output[(img > threshold[0]) & (img < threshold[1])]=1  
    return binary_output
```

```
In [11]: # testing sobel on test_image and warped image
```

```
Out[11]: Text(0.5,1,'Sobel Direction')
```



0.2 Combination of Color Transform and Gradients with example

The decision of choice of the shading spaces were arbitrary yet with a reason. I chose to utilize Saturation channel of HLS since it works kind of well under all conditions. Be that as it may, that was insufficient as it couldn't produce lines for specked white lines. I saw that the Lightness direct HLS functions admirably in all the conditions with the exception of the situation when the picture is excessively splendid. I chose to utilize and of both Saturation and Lightness Channel. Be that as it may, I was not even content with that as some black out edges were as yet not identified so I chose to utilize another luminance channel, this time from YUV colorspace-the Y channel.

When I was finished with choosing the shading space the following stage was to choose the Gradient I needed to apply. As should have been obvious clear vertical edges utilizing the x slope, I chose to utilize X angle as it were.

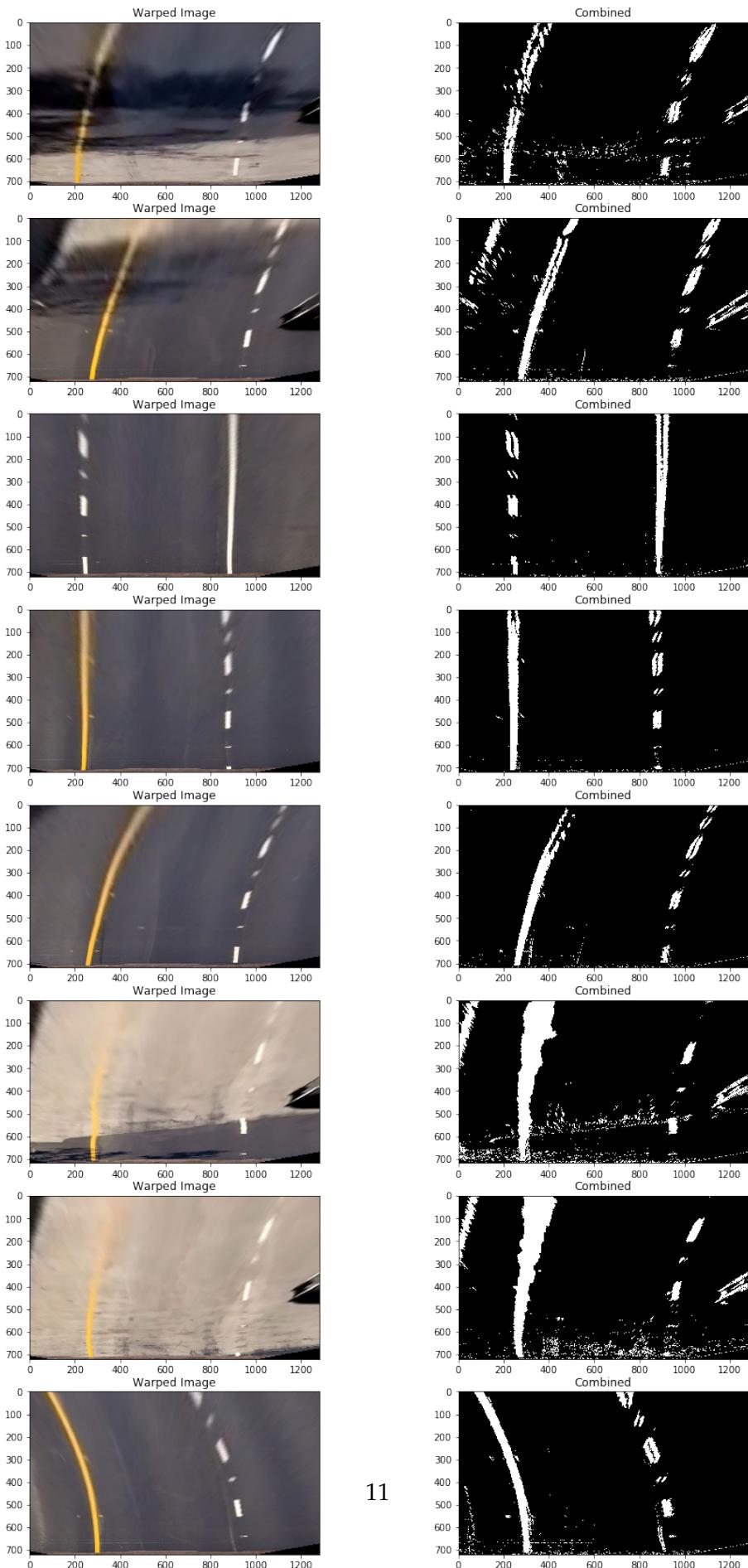
Last Combination-

1. Blend Channel 1 = Saturation Channel and Lightness Channel from HLS
2. Blend Channel 2 = Mix Channel 1 and Y channel for YUV
3. Last Combination= Mix Channel 2 or Sobel Gradient in X direction

In [12]: # Step 8- Combining Different ColorSpaces and Sobel Variants

```
def combineEverything(warpedImage, color_threshold, sobel_threshold):  
  
    s_channel = ExtractChannel(warpedImage, cv2.COLOR_RGB2HLS, color_threshold, 2)  
    l_channel = ExtractChannel(warpedImage, cv2.COLOR_RGB2HLS, color_threshold, 1)  
    y_channel = ExtractChannel(warpedImage, cv2.COLOR_RGB2YUV, color_threshold, 0)  
  
    sobelx = Sobel(warpedImage, sobel_threshold, 'x')  
    sobeldir = Sobel(warpedImage, [0.7, 25], 'dir')  
    #sobelxy=Sobel(warpedImage, sobel_threshold, 'xy')  
    combined_binary = np.zeros_like(s_channel)  
    combined_binary[((s_channel == 1) & (l_channel == 1)) & (y_channel == 1)] = (sobelx == 1)  
    return combined_binary
```

In [13]: #testing on test_images



0.3 Identifying lane-line pixels with Example

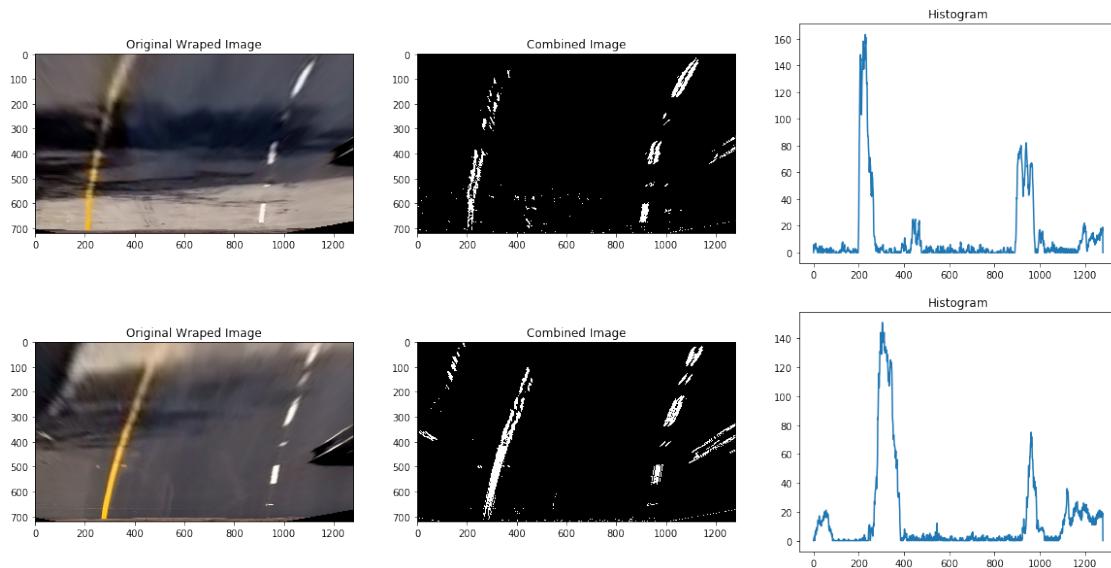
The first step is to create a Histogram of lower half of the image. With this way we are able to find out a distinction between the left lane pixels and right lane pixels.

In [14]: # Step 9 Plotting Histogram

```
def Histogram(warpedimage):
    return np.sum(warpedimage[warpedimage.shape[0]//2:,:,:], axis=0)
```

In [15]: # testing it on test images

Out [15]: Text(0.5,1,'Histogram')



The following stage is to start a Sliding Window Search in the left and right parts which we got from the histogram.

The sliding window is applied in following advances:

1. The left and right base focuses are determined from the histogram
2. We at that point figure the situation of all non zero x and non zero y pixels.
3. We at that point Start emphasizing over the windows where we start from focuses compute in point 1.
4. We at that point distinguish the non zero pixels in the window we simply characterized
5. We at that point gather all the lists in the rundown and choose the focal point of next window utilizing these focuses

6. When we are done, we separate the focuses to left and right positions
7. We at that point fit a second degree polynomial utilizing np.polyfit and point ascertain in sync

```
In [16]: # Step 10- Sliding Window Search
def SlidingWindowSearch(binary_warped, plot=False):

    histogram = Histogram(binary_warped)
    # Create an output image to draw on and visualize the result
    out_img = np.dstack((binary_warped, binary_warped, binary_warped))*255
    # Find the peak of the left and right halves of the histogram
    # These will be the starting point for the left and right lines
    midpoint = np.int(histogram.shape[0]/2)
    leftx_base = np.argmax(histogram[:midpoint])
    rightx_base = np.argmax(histogram[midpoint:]) + midpoint

    # Choose the number of sliding windows
    nwindows = 9
    # Set height of windows
    window_height = np.int(binary_warped.shape[0]/nwindows)
    # Identify the x and y positions of all nonzero pixels in the image
    nonzero = binary_warped.nonzero()
    nonzeroy = np.array(nonzero[0])
    nonzerox = np.array(nonzero[1])
    # Current positions to be updated for each window
    leftx_current = leftx_base
    rightx_current = rightx_base
    # Set the width of the windows +/- margin
    margin = 100
    # Set minimum number of pixels found to recenter window
    minpix = 50
    # Create empty lists to receive left and right lane pixel indices
    left_lane_inds = []
    right_lane_inds = []

    # Step through the windows one by one
    for window in range(nwindows):
        # Identify window boundaries in x and y (and right and left)
        win_y_low = binary_warped.shape[0] - (window+1)*window_height
        win_y_high = binary_warped.shape[0] - window*window_height
        win_xleft_low = leftx_current - margin
        win_xleft_high = leftx_current + margin
        win_xright_low = rightx_current - margin
        win_xright_high = rightx_current + margin
        # Draw the windows on the visualization image
        if(plot==True):
            cv2.rectangle(out_img,(win_xleft_low,win_y_low),(win_xleft_high,win_y_high),255,2)


```

```

        cv2.rectangle(out_img,(win_xright_low,win_y_low),(win_xright_high,win_y_high)
# Identify the nonzero pixels in x and y within the window
good_left_inds = ((nonzeroy >= win_y_low) & (nonzeroy < win_y_high) &
(nonzerox >= win_xleft_low) & (nonzerox < win_xleft_high)).nonzero()[0]
good_right_inds = ((nonzeroy >= win_y_low) & (nonzeroy < win_y_high) &
(nonzerox >= win_xright_low) & (nonzerox < win_xright_high)).nonzero()[0]
# Append these indices to the lists
left_lane_inds.append(good_left_inds)
right_lane_inds.append(good_right_inds)
# If you found > minpix pixels, recenter next window on their mean position
if len(good_left_inds) > minpix:
    leftx_current = np.int(np.mean(nonzerox[good_left_inds]))
if len(good_right_inds) > minpix:
    rightx_current = np.int(np.mean(nonzerox[good_right_inds]))

# Concatenate the arrays of indices
left_lane_inds = np.concatenate(left_lane_inds)
right_lane_inds = np.concatenate(right_lane_inds)

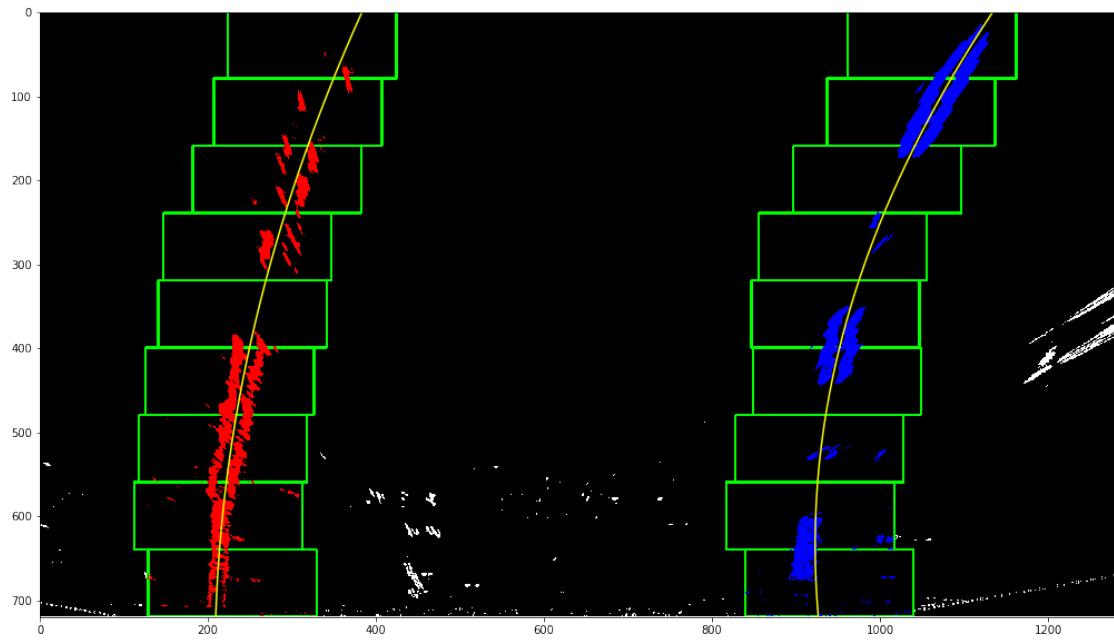
# Extract left and right line pixel positions
leftx = nonzerox[left_lane_inds]
lefty = nonzeroy[left_lane_inds]
rightx = nonzerox[right_lane_inds]
righty = nonzeroy[right_lane_inds]

# Fit a second order polynomial to each
left_fit = np.polyfit(lefty, leftx, 2)
right_fit = np.polyfit(righty, rightx, 2)

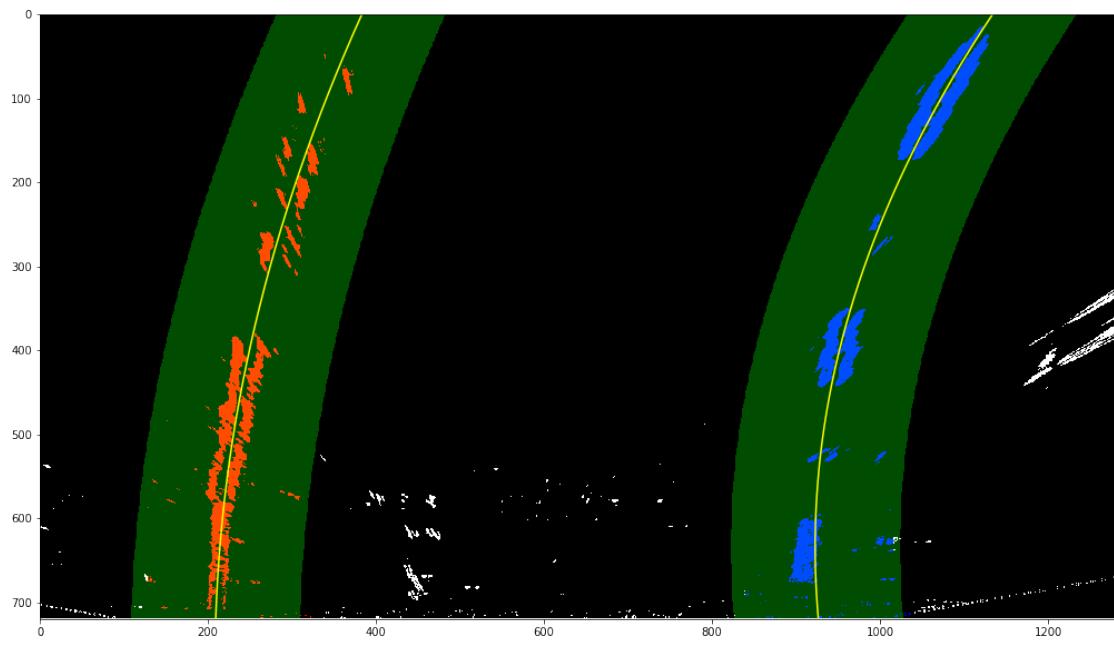
return left_fit,right_fit,left_lane_inds,right_lane_inds,out_img

```

In [17]: # Draw Sliding Window Visualization



```
In [18]: # Visualizing Smoothly
```



0.4 Radius of Curvature and Distance from Center Calculation

To compute Radius:-

1. First we characterize qualities to change over pixels to meters
2. Plot the left and right lines
3. Compute the ebb and flow from left and right paths seperately
4. Return mean of qualities determined in sync 3.

For Distance:-

We realize that the focal point of picture is the focal point of the vehicle. To ascertain the deviation from the middle, we can watch the pixel positions in the left path and the correct path. We take the mean of the left base most purpose of the left path and right base most purpose of the correct path and afterward take away it from the focal point of the vehicle to get the deviation from the inside.

```
In [21]: def CalculateRadiusOfCurvature(binary_warped, left_fit, right_fit):
    ym_per_pix = 30/720 # meters per pixel in y dimension
    xm_per_pix = 3.7/700 # meters per pixel in x dimension

    ploty = np.linspace(0, binary_warped.shape[0]-1, binary_warped.shape[0] )
    leftx = left_fit[0]*ploty**2 + left_fit[1]*ploty +left_fit[2]
    rightx = right_fit[0]*ploty**2 + right_fit[1]*ploty + right_fit[2]
    positionCar= binary_warped.shape[1]/2
    # Fit new polynomials to x,y in world space
    left_fit_cr = np.polyfit(ploty*ym_per_pix, leftx*xm_per_pix, 2)
    right_fit_cr = np.polyfit(ploty*ym_per_pix, rightx*xm_per_pix, 2)
    # Calculate the new radii of curvature

    y_eval=np.max(ploty)

    left_curverad = ((1 + (2*left_fit_cr[0]*y_eval*ym_per_pix + left_fit_cr[1])**2)**1.5)
    right_curverad = ((1 + (2*right_fit_cr[0]*y_eval*ym_per_pix + right_fit_cr[1])**2)**1.5)

    left_lane_bottom = (left_fit[0]*y_eval)**2 + left_fit[0]*y_eval + left_fit[2]
    right_lane_bottom = (right_fit[0]*y_eval)**2 + right_fit[0]*y_eval + right_fit[2]

    actualPosition= (left_lane_bottom+ right_lane_bottom)/2

    distance= (positionCar - actualPosition)* xm_per_pix

    # Now our radius of curvature is in meters
    #print(left_curverad, 'm', right_curverad, 'm')
    return (left_curverad + right_curverad)/2, distance
    # Example values: 632.1 m      626.2 m
```

0.5 Image Plotted Back with Example

When we are finished with this the subsequent stage is to unwarp the picture back to unique picture. To do so following advances were followed:-

1. Recast the x and y point to give as contribution to cv2.fillPoly. These are similar focuses we got from fitting the lines.
2. Ascertain the Minv which is Inverse Matrix. This is finished by passing the opposite focuses this chance to getPerspectiveTransform work
3. Draw the sidelines from the focuses chose in sync 1 onto a clear twisted picture
4. Unwarp the picture utilizing cv2.warpPerspective.
5. Consolidate the first picture with the picture we got from stage 4 to plot the path lines.

In [19]: # Unwarp Image and plot line

```
def DrawLine(original_image,binary_warped, left_fit, right_fit):

    h,w= binary_warped.shape
    Minv = cv2.getPerspectiveTransform(dst, src)
    ploty = np.linspace(0, binary_warped.shape[0]-1, binary_warped.shape[0] )
    left_fitx = left_fit[0]*ploty**2 + left_fit[1]*ploty +left_fit[2]
    right_fitx = right_fit[0]*ploty**2 + right_fit[1]*ploty + right_fit[2]
    warp_zero = np.zeros_like(binary_warped).astype(np.uint8)
    color_warp = np.dstack((warp_zero, warp_zero, warp_zero))

    ploty = np.linspace(0, h-1, num=h)# to cover same y-range as image

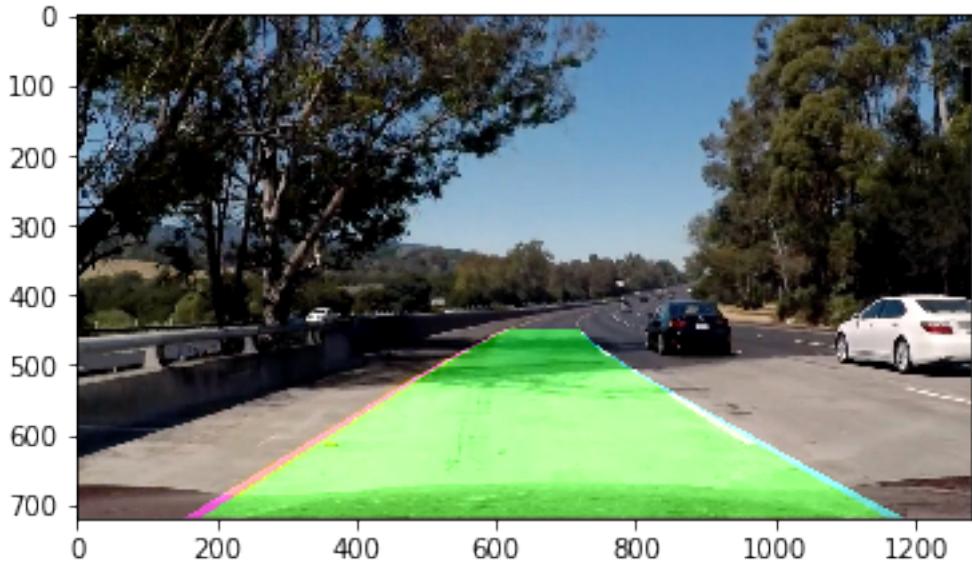
    # Recast the x and y points into usable format for cv2.fillPoly()
    pts_left = np.array([np.transpose(np.vstack([left_fitx, ploty]))])
    pts_right = np.array([np.flipud(np.transpose(np.vstack([right_fitx, ploty])))])
    pts = np.hstack((pts_left, pts_right))

    # Draw the lane onto the warped blank image
    cv2.fillPoly(color_warp, np.int_(pts), (0,255, 0))
    cv2.polyline(color_warp, np.int32([pts_left]), isClosed=False, color=(255,0,255),
    cv2.polyline(color_warp, np.int32([pts_right]), isClosed=False, color=(0,255,255),

    # Warp the blank back to original image space using inverse perspective matrix (Minv)
    newwarp = cv2.warpPerspective(color_warp, Minv, (w, h))
    #axes[index+1].imshow(newwarp)
    # Combine the result with the original image
    result = cv2.addWeighted(original_image, 1, newwarp, 0.5, 0)
    return result
```

In [20]: # testing it on test image

Out[20]: <matplotlib.image.AxesImage at 0x7f788bc5de80>



0.5.1 Defining Pipeline

```
In [22]: from random import randint
import datetime
import time
def pipeline(originalImage):
    originalImage= cv2.cvtColor(originalImage, cv2.COLOR_BGR2RGB)
    undistortedImage= undistortImage(originalImage)
    warpedImage= WarpPerspective(undistortedImage)
    combinedImage= combineEverything(warpedImage,color_threshold= [100,255],sobel_thres
    returnedOutput = SlidingWindowSearch(combinedImage)
    left_fit=returnedOutput[0]
    right_fit=returnedOutput[1]
    #VisualizeSlidingWindow(combinedImage, left_fit,right_fit, returnedOutput[2], return
    finalImage=DrawLine(originalImage,combinedImage,left_fit,right_fit)
    #cv2.imwrite('./test/'+str(randint(0, 99999))+'.jpg',originalImage)

    radius, distance = CalculateRadiusOfCurvature(combinedImage,left_fit,right_fit)
    cv2.putText(finalImage,"Radius of Curvature is " + str(int(radius))+ "m", (100,100)
    #print(distance)
    cv2.putText(finalImage,"Distance from center is {:.2f}m".format(distance)+ "m", (100,
    ts = time.time()
    st = datetime.datetime.fromtimestamp(ts).strftime('%Y%m%d %H%M%S')

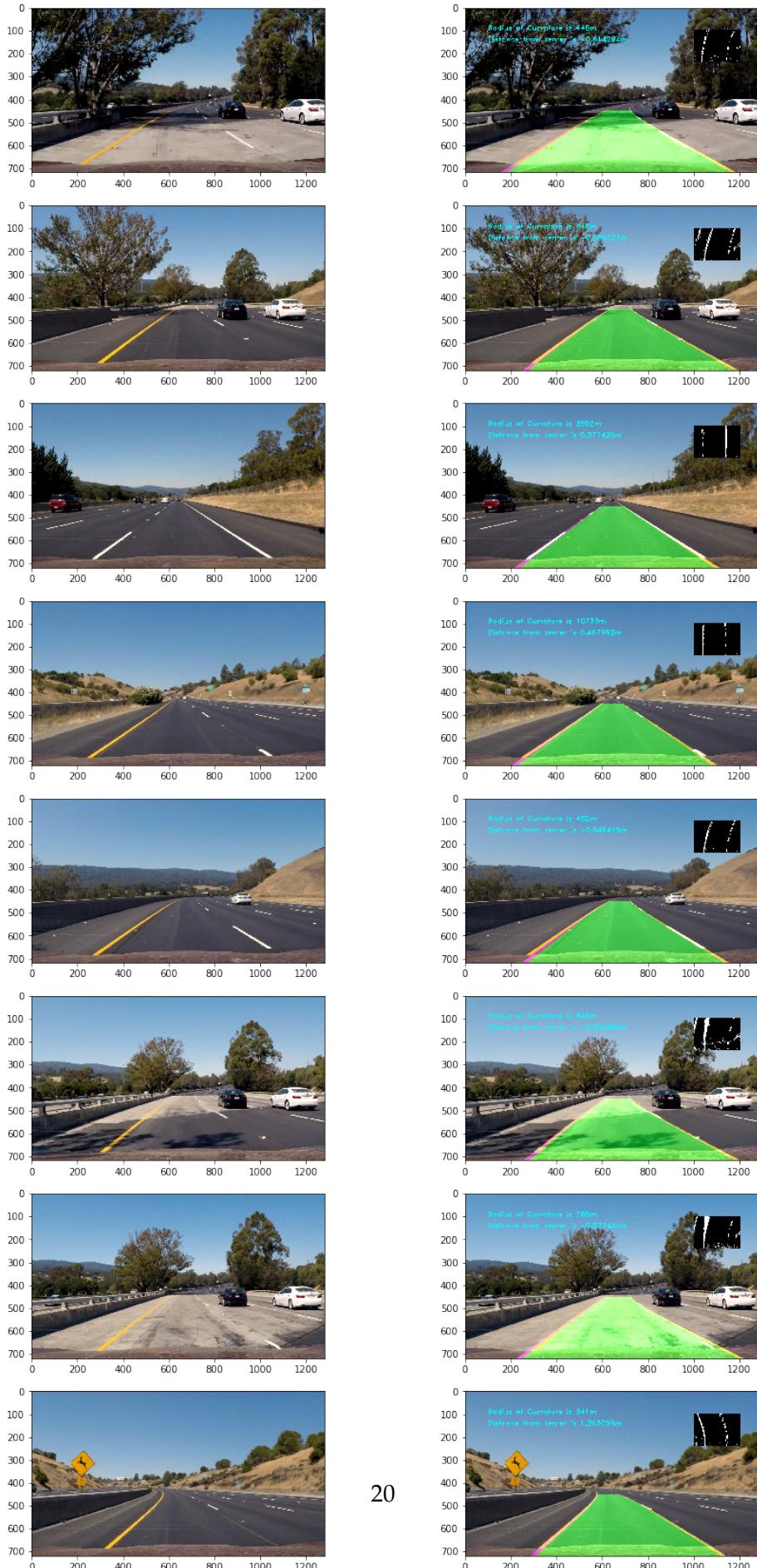
    cv2.imwrite('./Output_1/'+str(st)+'.jpg',originalImage)

    cv2.imwrite('./Output_1/'+str(st)+'_o.jpg',finalImage)
    newCombinedImage= np.dstack((combinedImage*255,combinedImage*255,combinedImage*255))
```

```
finalImage[100:240,1000:1200, :] = cv2.resize(newCombinedImage, (200,140))
return cv2.cvtColor(finalImage, cv2.COLOR_BGR2RGB)
```

0.5.2 Running Pipeline on Test Images

In [23]: *#testing on test images*



0.5.3 Step 15- Running Pipeline on Test Video

```
In [27]: import moviepy
        from moviepy.editor import VideoFileClip
        video_output1 = 'project_video_output.mp4'
        video_input1 = VideoFileClip('project_video.mp4')
        processed_video = video_input1.fl_image(pipeline)
        %time processed_video.write_videofile(video_output1, audio=False)
```

```
[MoviePy] >>> Building video project_video_output.mp4
[MoviePy] Writing video project_video_output.mp4
```

```
0% | 0/1261 [00:00<?, ?it/s]
0% | 1/1261 [00:00<06:01, 3.49it/s]
0% | 2/1261 [00:00<06:07, 3.43it/s]
0% | 3/1261 [00:00<06:02, 3.47it/s]
0% | 4/1261 [00:01<05:56, 3.53it/s]
0% | 5/1261 [00:01<05:53, 3.56it/s]
0% | 6/1261 [00:01<05:52, 3.56it/s]
1% | 7/1261 [00:01<05:49, 3.59it/s]
1% | 8/1261 [00:02<05:47, 3.60it/s]
1% | 9/1261 [00:02<05:48, 3.60it/s]
1% | 10/1261 [00:02<05:45, 3.62it/s]
1% | 11/1261 [00:03<05:45, 3.62it/s]
1% | 12/1261 [00:03<05:45, 3.62it/s]
1% | 13/1261 [00:03<05:44, 3.62it/s]
1% | 14/1261 [00:03<05:44, 3.62it/s]
1% | 15/1261 [00:04<05:44, 3.62it/s]
1% | 16/1261 [00:04<05:43, 3.62it/s]
1% | 17/1261 [00:04<05:44, 3.61it/s]
1% | 18/1261 [00:05<05:45, 3.60it/s]
2% | 19/1261 [00:05<05:43, 3.62it/s]
2% | 20/1261 [00:05<05:44, 3.60it/s]
2% | 21/1261 [00:05<05:44, 3.59it/s]
2% | 22/1261 [00:06<05:43, 3.60it/s]
2% | 23/1261 [00:06<05:41, 3.62it/s]
2% | 24/1261 [00:06<05:41, 3.62it/s]
2% | 25/1261 [00:06<05:41, 3.62it/s]
2% | 26/1261 [00:07<05:40, 3.62it/s]
2% | 27/1261 [00:07<05:39, 3.64it/s]
2% | 28/1261 [00:07<05:37, 3.65it/s]
2% | 29/1261 [00:08<05:36, 3.66it/s]
2% | 30/1261 [00:08<05:38, 3.63it/s]
```

2%	31/1261 [00:08<05:38,	3.63it/s]
3%	32/1261 [00:08<05:37,	3.64it/s]
3%	33/1261 [00:09<05:36,	3.65it/s]
3%	34/1261 [00:09<05:36,	3.65it/s]
3%	35/1261 [00:09<05:37,	3.64it/s]
3%	36/1261 [00:09<05:35,	3.65it/s]
3%	37/1261 [00:10<05:35,	3.64it/s]
3%	38/1261 [00:10<05:36,	3.64it/s]
3%	39/1261 [00:10<05:35,	3.65it/s]
3%	40/1261 [00:11<05:35,	3.64it/s]
3%	41/1261 [00:11<05:34,	3.65it/s]
3%	42/1261 [00:11<05:33,	3.66it/s]
3%	43/1261 [00:12<07:14,	2.80it/s]
3%	44/1261 [00:12<07:11,	2.82it/s]
4%	45/1261 [00:12<07:13,	2.80it/s]
4%	46/1261 [00:13<07:15,	2.79it/s]
4%	47/1261 [00:13<07:09,	2.83it/s]
4%	48/1261 [00:13<07:07,	2.84it/s]
4%	49/1261 [00:14<07:11,	2.81it/s]
4%	50/1261 [00:14<07:02,	2.87it/s]
4%	51/1261 [00:14<07:10,	2.81it/s]
4%	52/1261 [00:15<07:01,	2.87it/s]
4%	53/1261 [00:15<07:10,	2.80it/s]
4%	54/1261 [00:16<07:04,	2.84it/s]
4%	55/1261 [00:16<07:15,	2.77it/s]
4%	56/1261 [00:16<07:03,	2.85it/s]
5%	57/1261 [00:17<07:13,	2.78it/s]
5%	58/1261 [00:17<07:02,	2.84it/s]
5%	59/1261 [00:17<07:14,	2.77it/s]
5%	60/1261 [00:18<07:07,	2.81it/s]
5%	61/1261 [00:18<07:11,	2.78it/s]
5%	62/1261 [00:18<07:17,	2.74it/s]
5%	63/1261 [00:19<07:06,	2.81it/s]
5%	64/1261 [00:19<07:16,	2.74it/s]
5%	65/1261 [00:19<07:05,	2.81it/s]
5%	66/1261 [00:20<07:13,	2.75it/s]
5%	67/1261 [00:20<07:05,	2.81it/s]
5%	68/1261 [00:21<07:13,	2.75it/s]
5%	69/1261 [00:21<07:03,	2.82it/s]
6%	70/1261 [00:21<07:11,	2.76it/s]
6%	71/1261 [00:22<07:01,	2.82it/s]
6%	72/1261 [00:22<07:06,	2.79it/s]
6%	73/1261 [00:22<07:15,	2.73it/s]
6%	74/1261 [00:23<07:03,	2.80it/s]
6%	75/1261 [00:23<07:10,	2.76it/s]
6%	76/1261 [00:23<06:58,	2.83it/s]
6%	77/1261 [00:24<07:08,	2.77it/s]
6%	78/1261 [00:24<06:58,	2.83it/s]

6%	79/1261 [00:25<07:08, 2.76it/s]
6%	80/1261 [00:25<06:57, 2.83it/s]
6%	81/1261 [00:25<07:05, 2.78it/s]
7%	82/1261 [00:26<06:56, 2.83it/s]
7%	83/1261 [00:26<06:58, 2.82it/s]
7%	84/1261 [00:26<07:02, 2.79it/s]
7%	85/1261 [00:27<06:51, 2.86it/s]
7%	86/1261 [00:27<06:57, 2.82it/s]
7%	87/1261 [00:27<06:59, 2.80it/s]
7%	88/1261 [00:28<07:03, 2.77it/s]
7%	89/1261 [00:28<06:54, 2.83it/s]
7%	90/1261 [00:28<07:04, 2.76it/s]
7%	91/1261 [00:29<06:56, 2.81it/s]
7%	92/1261 [00:29<07:05, 2.75it/s]
7%	93/1261 [00:30<06:54, 2.82it/s]
7%	94/1261 [00:30<06:59, 2.78it/s]
8%	95/1261 [00:30<06:47, 2.86it/s]
8%	96/1261 [00:31<06:55, 2.80it/s]
8%	97/1261 [00:31<06:48, 2.85it/s]
8%	98/1261 [00:31<06:53, 2.81it/s]
8%	99/1261 [00:32<06:47, 2.85it/s]
8%	100/1261 [00:32<06:52, 2.82it/s]
8%	101/1261 [00:32<06:53, 2.81it/s]
8%	102/1261 [00:33<06:59, 2.76it/s]
8%	103/1261 [00:33<06:50, 2.82it/s]
8%	104/1261 [00:33<06:52, 2.81it/s]
8%	105/1261 [00:34<07:37, 2.53it/s]
8%	106/1261 [00:34<07:14, 2.66it/s]
8%	107/1261 [00:35<07:14, 2.66it/s]
9%	108/1261 [00:35<07:00, 2.75it/s]
9%	109/1261 [00:35<07:02, 2.73it/s]
9%	110/1261 [00:36<06:52, 2.79it/s]
9%	111/1261 [00:36<06:57, 2.76it/s]
9%	112/1261 [00:36<06:48, 2.81it/s]
9%	113/1261 [00:37<06:55, 2.76it/s]
9%	114/1261 [00:37<06:46, 2.82it/s]
9%	115/1261 [00:37<06:50, 2.79it/s]
9%	116/1261 [00:38<06:41, 2.85it/s]
9%	117/1261 [00:38<06:48, 2.80it/s]
9%	118/1261 [00:39<06:40, 2.85it/s]
9%	119/1261 [00:39<06:49, 2.79it/s]
10%	120/1261 [00:39<06:43, 2.83it/s]
10%	121/1261 [00:40<06:49, 2.78it/s]
10%	122/1261 [00:40<06:53, 2.75it/s]
10%	123/1261 [00:40<06:44, 2.82it/s]
10%	124/1261 [00:41<06:52, 2.76it/s]
10%	125/1261 [00:41<06:41, 2.83it/s]
10%	126/1261 [00:41<06:47, 2.79it/s]

10%	127/1261 [00:42<06:37, 2.85it/s]
10%	128/1261 [00:42<06:47, 2.78it/s]
10%	129/1261 [00:42<06:39, 2.84it/s]
10%	130/1261 [00:43<06:45, 2.79it/s]
10%	131/1261 [00:43<06:36, 2.85it/s]
10%	132/1261 [00:44<06:45, 2.78it/s]
11%	133/1261 [00:44<06:38, 2.83it/s]
11%	134/1261 [00:44<06:43, 2.79it/s]
11%	135/1261 [00:45<06:47, 2.76it/s]
11%	136/1261 [00:45<06:38, 2.82it/s]
11%	137/1261 [00:45<06:50, 2.74it/s]
11%	138/1261 [00:46<06:37, 2.82it/s]
11%	139/1261 [00:46<06:44, 2.77it/s]
11%	140/1261 [00:46<06:34, 2.84it/s]
11%	141/1261 [00:47<06:41, 2.79it/s]
11%	142/1261 [00:47<06:34, 2.84it/s]
11%	143/1261 [00:47<06:40, 2.79it/s]
11%	144/1261 [00:48<06:46, 2.75it/s]
11%	145/1261 [00:48<06:36, 2.81it/s]
12%	146/1261 [00:49<06:43, 2.77it/s]
12%	147/1261 [00:49<06:34, 2.83it/s]
12%	148/1261 [00:49<06:43, 2.76it/s]
12%	149/1261 [00:50<06:34, 2.82it/s]
12%	150/1261 [00:50<06:45, 2.74it/s]
12%	151/1261 [00:50<06:36, 2.80it/s]
12%	152/1261 [00:51<06:43, 2.75it/s]
12%	153/1261 [00:51<06:34, 2.81it/s]
12%	154/1261 [00:51<06:40, 2.76it/s]
12%	155/1261 [00:52<06:30, 2.83it/s]
12%	156/1261 [00:52<06:34, 2.80it/s]
12%	157/1261 [00:52<06:26, 2.86it/s]
13%	158/1261 [00:53<06:34, 2.80it/s]
13%	159/1261 [00:53<06:26, 2.85it/s]
13%	160/1261 [00:54<06:33, 2.80it/s]
13%	161/1261 [00:54<06:24, 2.86it/s]
13%	162/1261 [00:54<06:35, 2.78it/s]
13%	163/1261 [00:55<08:18, 2.20it/s]
13%	164/1261 [00:55<07:53, 2.32it/s]
13%	165/1261 [00:56<07:20, 2.49it/s]
13%	166/1261 [00:56<07:10, 2.55it/s]
13%	167/1261 [00:56<06:51, 2.66it/s]
13%	168/1261 [00:57<06:50, 2.66it/s]
13%	169/1261 [00:57<06:34, 2.77it/s]
13%	170/1261 [00:57<06:39, 2.73it/s]
14%	171/1261 [00:58<06:26, 2.82it/s]
14%	172/1261 [00:58<06:30, 2.79it/s]
14%	173/1261 [00:58<06:21, 2.85it/s]
14%	174/1261 [00:59<06:28, 2.80it/s]

14%	175/1261	[00:59<06:22,	2.84it/s]
14%	176/1261	[01:00<06:25,	2.81it/s]
14%	177/1261	[01:00<06:30,	2.77it/s]
14%	178/1261	[01:00<06:23,	2.82it/s]
14%	179/1261	[01:01<06:30,	2.77it/s]
14%	180/1261	[01:01<06:23,	2.82it/s]
14%	181/1261	[01:01<06:31,	2.76it/s]
14%	182/1261	[01:02<06:21,	2.83it/s]
15%	183/1261	[01:02<06:24,	2.81it/s]
15%	184/1261	[01:02<06:26,	2.79it/s]
15%	185/1261	[01:03<06:32,	2.74it/s]
15%	186/1261	[01:03<06:23,	2.81it/s]
15%	187/1261	[01:03<06:25,	2.79it/s]
15%	188/1261	[01:04<06:30,	2.75it/s]
15%	189/1261	[01:04<06:22,	2.80it/s]
15%	190/1261	[01:05<06:24,	2.79it/s]
15%	191/1261	[01:05<06:28,	2.76it/s]
15%	192/1261	[01:05<06:20,	2.81it/s]
15%	193/1261	[01:06<06:29,	2.74it/s]
15%	194/1261	[01:06<07:01,	2.53it/s]
15%	195/1261	[01:06<06:54,	2.57it/s]
16%	196/1261	[01:07<06:36,	2.68it/s]
16%	197/1261	[01:07<06:38,	2.67it/s]
16%	198/1261	[01:08<06:26,	2.75it/s]
16%	199/1261	[01:08<06:29,	2.73it/s]
16%	200/1261	[01:08<06:18,	2.80it/s]
16%	201/1261	[01:09<06:22,	2.77it/s]
16%	202/1261	[01:09<06:11,	2.85it/s]
16%	203/1261	[01:09<06:18,	2.80it/s]
16%	204/1261	[01:10<06:08,	2.87it/s]
16%	205/1261	[01:10<06:15,	2.81it/s]
16%	206/1261	[01:10<06:08,	2.86it/s]
16%	207/1261	[01:11<06:18,	2.79it/s]
16%	208/1261	[01:11<06:10,	2.84it/s]
17%	209/1261	[01:11<06:21,	2.76it/s]
17%	210/1261	[01:12<06:15,	2.80it/s]
17%	211/1261	[01:12<06:20,	2.76it/s]
17%	212/1261	[01:13<06:10,	2.83it/s]
17%	213/1261	[01:13<06:17,	2.78it/s]
17%	214/1261	[01:13<06:07,	2.85it/s]
17%	215/1261	[01:14<06:13,	2.80it/s]
17%	216/1261	[01:14<06:03,	2.87it/s]
17%	217/1261	[01:14<06:11,	2.81it/s]
17%	218/1261	[01:15<06:02,	2.88it/s]
17%	219/1261	[01:15<06:12,	2.80it/s]
17%	220/1261	[01:15<06:03,	2.86it/s]
18%	221/1261	[01:16<06:11,	2.80it/s]
18%	222/1261	[01:16<06:03,	2.86it/s]

18%	223/1261 [01:16<06:15, 2.77it/s]
18%	224/1261 [01:17<06:08, 2.82it/s]
18%	225/1261 [01:17<06:16, 2.75it/s]
18%	226/1261 [01:17<06:09, 2.80it/s]
18%	227/1261 [01:18<06:12, 2.78it/s]
18%	228/1261 [01:18<06:03, 2.84it/s]
18%	229/1261 [01:19<06:10, 2.79it/s]
18%	230/1261 [01:19<06:02, 2.84it/s]
18%	231/1261 [01:19<06:10, 2.78it/s]
18%	232/1261 [01:20<06:01, 2.85it/s]
18%	233/1261 [01:20<06:09, 2.78it/s]
19%	234/1261 [01:20<06:02, 2.84it/s]
19%	235/1261 [01:21<06:08, 2.78it/s]
19%	236/1261 [01:21<06:01, 2.84it/s]
19%	237/1261 [01:21<06:03, 2.82it/s]
19%	238/1261 [01:22<06:09, 2.77it/s]
19%	239/1261 [01:22<06:01, 2.83it/s]
19%	240/1261 [01:22<06:12, 2.74it/s]
19%	241/1261 [01:23<06:02, 2.82it/s]
19%	242/1261 [01:23<06:07, 2.78it/s]
19%	243/1261 [01:24<05:56, 2.85it/s]
19%	244/1261 [01:24<06:00, 2.82it/s]
19%	245/1261 [01:24<06:04, 2.79it/s]
20%	246/1261 [01:25<05:55, 2.85it/s]
20%	247/1261 [01:25<06:00, 2.81it/s]
20%	248/1261 [01:25<06:05, 2.77it/s]
20%	249/1261 [01:26<05:56, 2.84it/s]
20%	250/1261 [01:26<06:03, 2.78it/s]
20%	251/1261 [01:26<05:55, 2.84it/s]
20%	252/1261 [01:27<06:06, 2.75it/s]
20%	253/1261 [01:27<05:57, 2.82it/s]
20%	254/1261 [01:27<06:01, 2.79it/s]
20%	255/1261 [01:28<05:52, 2.85it/s]
20%	256/1261 [01:28<05:51, 2.86it/s]
20%	257/1261 [01:29<05:59, 2.80it/s]
20%	258/1261 [01:29<05:52, 2.85it/s]
21%	259/1261 [01:29<05:59, 2.79it/s]
21%	260/1261 [01:30<05:52, 2.84it/s]
21%	261/1261 [01:30<06:00, 2.78it/s]
21%	262/1261 [01:30<05:53, 2.82it/s]
21%	263/1261 [01:31<06:01, 2.76it/s]
21%	264/1261 [01:31<05:53, 2.82it/s]
21%	265/1261 [01:31<06:00, 2.76it/s]
21%	266/1261 [01:32<05:51, 2.83it/s]
21%	267/1261 [01:32<05:58, 2.77it/s]
21%	268/1261 [01:32<05:51, 2.83it/s]
21%	269/1261 [01:33<05:58, 2.77it/s]
21%	270/1261 [01:33<05:53, 2.81it/s]

21%		271/1261	[01:34<05:56,	2.78it/s]
22%		272/1261	[01:34<06:00,	2.74it/s]
22%		273/1261	[01:34<05:51,	2.81it/s]
22%		274/1261	[01:35<06:00,	2.74it/s]
22%		275/1261	[01:35<05:49,	2.82it/s]
22%		276/1261	[01:35<05:54,	2.78it/s]
22%		277/1261	[01:36<05:44,	2.86it/s]
22%		278/1261	[01:36<05:48,	2.82it/s]
22%		279/1261	[01:36<05:40,	2.89it/s]
22%		280/1261	[01:37<05:46,	2.83it/s]
22%		281/1261	[01:37<05:38,	2.90it/s]
22%		282/1261	[01:37<05:50,	2.79it/s]
22%		283/1261	[01:38<05:42,	2.85it/s]
23%		284/1261	[01:38<05:45,	2.83it/s]
23%		285/1261	[01:38<05:43,	2.84it/s]
23%		286/1261	[01:39<05:41,	2.85it/s]
23%		287/1261	[01:39<05:39,	2.87it/s]
23%		288/1261	[01:40<05:42,	2.84it/s]
23%		289/1261	[01:40<05:44,	2.82it/s]
23%		290/1261	[01:40<05:37,	2.88it/s]
23%		291/1261	[01:41<05:44,	2.82it/s]
23%		292/1261	[01:41<05:38,	2.86it/s]
23%		293/1261	[01:41<06:04,	2.66it/s]
23%		294/1261	[01:42<05:53,	2.74it/s]
23%		295/1261	[01:42<05:37,	2.86it/s]
23%		296/1261	[01:42<05:41,	2.83it/s]
24%		297/1261	[01:43<05:33,	2.89it/s]
24%		298/1261	[01:43<05:40,	2.83it/s]
24%		299/1261	[01:43<05:34,	2.88it/s]
24%		300/1261	[01:44<05:44,	2.79it/s]
24%		301/1261	[01:44<05:36,	2.85it/s]
24%		302/1261	[01:45<05:44,	2.78it/s]
24%		303/1261	[01:45<05:37,	2.84it/s]
24%		304/1261	[01:45<05:42,	2.79it/s]
24%		305/1261	[01:46<05:34,	2.85it/s]
24%		306/1261	[01:46<05:40,	2.80it/s]
24%		307/1261	[01:46<05:34,	2.85it/s]
24%		308/1261	[01:47<05:40,	2.80it/s]
25%		309/1261	[01:47<05:32,	2.86it/s]
25%		310/1261	[01:47<05:38,	2.81it/s]
25%		311/1261	[01:48<05:32,	2.86it/s]
25%		312/1261	[01:48<05:41,	2.78it/s]
25%		313/1261	[01:48<05:33,	2.84it/s]
25%		314/1261	[01:49<05:40,	2.78it/s]
25%		315/1261	[01:49<05:32,	2.85it/s]
25%		316/1261	[01:49<05:38,	2.79it/s]
25%		317/1261	[01:50<05:32,	2.84it/s]
25%		318/1261	[01:50<05:38,	2.79it/s]

25%		319/1261	[01:51<05:31,	2.84it/s]
25%		320/1261	[01:51<05:39,	2.77it/s]
25%		321/1261	[01:51<05:29,	2.85it/s]
26%		322/1261	[01:52<05:37,	2.78it/s]
26%		323/1261	[01:52<05:32,	2.82it/s]
26%		324/1261	[01:52<05:27,	2.86it/s]
26%		325/1261	[01:53<05:37,	2.77it/s]
26%		326/1261	[01:53<05:28,	2.85it/s]
26%		327/1261	[01:53<05:33,	2.80it/s]
26%		328/1261	[01:54<05:27,	2.85it/s]
26%		329/1261	[01:54<05:30,	2.82it/s]
26%		330/1261	[01:54<05:25,	2.86it/s]
26%		331/1261	[01:55<05:34,	2.78it/s]
26%		332/1261	[01:55<05:27,	2.84it/s]
26%		333/1261	[01:56<05:35,	2.76it/s]
26%		334/1261	[01:56<05:27,	2.83it/s]
27%		335/1261	[01:56<05:34,	2.77it/s]
27%		336/1261	[01:57<05:28,	2.82it/s]
27%		337/1261	[01:57<05:34,	2.76it/s]
27%		338/1261	[01:57<05:27,	2.82it/s]
27%		339/1261	[01:58<05:31,	2.78it/s]
27%		340/1261	[01:58<05:24,	2.84it/s]
27%		341/1261	[01:58<05:29,	2.79it/s]
27%		342/1261	[01:59<05:22,	2.85it/s]
27%		343/1261	[01:59<05:32,	2.76it/s]
27%		344/1261	[01:59<05:24,	2.82it/s]
27%		345/1261	[02:00<05:29,	2.78it/s]
27%		346/1261	[02:00<05:20,	2.86it/s]
28%		347/1261	[02:00<05:24,	2.81it/s]
28%		348/1261	[02:01<05:16,	2.88it/s]
28%		349/1261	[02:01<05:23,	2.82it/s]
28%		350/1261	[02:02<05:17,	2.87it/s]
28%		351/1261	[02:02<05:22,	2.82it/s]
28%		352/1261	[02:02<05:16,	2.87it/s]
28%		353/1261	[02:03<05:21,	2.83it/s]
28%		354/1261	[02:03<05:15,	2.88it/s]
28%		355/1261	[02:03<05:23,	2.80it/s]
28%		356/1261	[02:04<05:18,	2.84it/s]
28%		357/1261	[02:04<05:15,	2.87it/s]
28%		358/1261	[02:04<05:26,	2.77it/s]
28%		359/1261	[02:05<05:18,	2.84it/s]
29%		360/1261	[02:05<05:26,	2.76it/s]
29%		361/1261	[02:05<05:19,	2.81it/s]
29%		362/1261	[02:06<05:26,	2.76it/s]
29%		363/1261	[02:06<05:17,	2.83it/s]
29%		364/1261	[02:07<05:22,	2.78it/s]
29%		365/1261	[02:07<05:13,	2.86it/s]
29%		366/1261	[02:07<05:20,	2.80it/s]

29%	367/1261 [02:08<05:12, 2.86it/s]
29%	368/1261 [02:08<05:18, 2.80it/s]
29%	369/1261 [02:08<05:11, 2.86it/s]
29%	370/1261 [02:09<05:21, 2.77it/s]
29%	371/1261 [02:09<05:14, 2.83it/s]
30%	372/1261 [02:09<05:16, 2.81it/s]
30%	373/1261 [02:10<05:23, 2.74it/s]
30%	374/1261 [02:10<05:15, 2.81it/s]
30%	375/1261 [02:10<05:16, 2.80it/s]
30%	376/1261 [02:11<05:21, 2.76it/s]
30%	377/1261 [02:11<05:13, 2.82it/s]
30%	378/1261 [02:12<05:18, 2.77it/s]
30%	379/1261 [02:12<05:09, 2.85it/s]
30%	380/1261 [02:12<05:10, 2.84it/s]
30%	381/1261 [02:13<05:15, 2.79it/s]
30%	382/1261 [02:13<05:06, 2.86it/s]
30%	383/1261 [02:13<05:13, 2.80it/s]
30%	384/1261 [02:14<05:06, 2.86it/s]
31%	385/1261 [02:14<05:13, 2.80it/s]
31%	386/1261 [02:14<05:06, 2.86it/s]
31%	387/1261 [02:15<05:12, 2.80it/s]
31%	388/1261 [02:15<05:04, 2.87it/s]
31%	389/1261 [02:15<05:12, 2.79it/s]
31%	390/1261 [02:16<05:07, 2.84it/s]
31%	391/1261 [02:16<05:15, 2.76it/s]
31%	392/1261 [02:16<05:08, 2.81it/s]
31%	393/1261 [02:17<05:12, 2.77it/s]
31%	394/1261 [02:17<05:04, 2.85it/s]
31%	395/1261 [02:18<05:10, 2.79it/s]
31%	396/1261 [02:18<05:03, 2.85it/s]
31%	397/1261 [02:18<05:09, 2.79it/s]
32%	398/1261 [02:19<05:01, 2.86it/s]
32%	399/1261 [02:19<05:05, 2.82it/s]
32%	400/1261 [02:19<05:00, 2.86it/s]
32%	401/1261 [02:20<05:04, 2.83it/s]
32%	402/1261 [02:20<04:58, 2.88it/s]
32%	403/1261 [02:20<05:07, 2.79it/s]
32%	404/1261 [02:21<05:01, 2.84it/s]
32%	405/1261 [02:21<05:05, 2.81it/s]
32%	406/1261 [02:21<05:07, 2.78it/s]
32%	407/1261 [02:22<04:59, 2.85it/s]
32%	408/1261 [02:22<05:04, 2.80it/s]
32%	409/1261 [02:22<04:57, 2.86it/s]
33%	410/1261 [02:23<05:03, 2.81it/s]
33%	411/1261 [02:23<04:54, 2.89it/s]
33%	412/1261 [02:24<05:00, 2.82it/s]
33%	413/1261 [02:24<04:52, 2.90it/s]
33%	414/1261 [02:24<05:02, 2.80it/s]

33%	415/1261 [02:25<04:57, 2.85it/s]
33%	416/1261 [02:25<05:03, 2.78it/s]
33%	417/1261 [02:25<04:55, 2.86it/s]
33%	418/1261 [02:26<05:03, 2.78it/s]
33%	419/1261 [02:26<04:56, 2.84it/s]
33%	420/1261 [02:26<05:03, 2.77it/s]
33%	421/1261 [02:27<04:56, 2.84it/s]
33%	422/1261 [02:27<05:07, 2.73it/s]
34%	423/1261 [02:27<05:10, 2.70it/s]
34%	424/1261 [02:28<05:00, 2.78it/s]
34%	425/1261 [02:28<05:05, 2.74it/s]
34%	426/1261 [02:29<04:55, 2.82it/s]
34%	427/1261 [02:29<04:58, 2.79it/s]
34%	428/1261 [02:29<04:51, 2.85it/s]
34%	429/1261 [02:30<04:58, 2.78it/s]
34%	430/1261 [02:30<04:51, 2.85it/s]
34%	431/1261 [02:30<04:57, 2.79it/s]
34%	432/1261 [02:31<04:49, 2.86it/s]
34%	433/1261 [02:31<04:58, 2.77it/s]
34%	434/1261 [02:31<04:52, 2.83it/s]
34%	435/1261 [02:32<04:57, 2.78it/s]
35%	436/1261 [02:32<04:50, 2.84it/s]
35%	437/1261 [02:32<04:53, 2.80it/s]
35%	438/1261 [02:33<04:46, 2.87it/s]
35%	439/1261 [02:33<04:52, 2.81it/s]
35%	440/1261 [02:33<04:46, 2.86it/s]
35%	441/1261 [02:34<04:52, 2.80it/s]
35%	442/1261 [02:34<04:46, 2.86it/s]
35%	443/1261 [02:35<04:48, 2.83it/s]
35%	444/1261 [02:35<04:52, 2.79it/s]
35%	445/1261 [02:35<04:46, 2.85it/s]
35%	446/1261 [02:36<04:49, 2.81it/s]
35%	447/1261 [02:36<04:43, 2.87it/s]
36%	448/1261 [02:36<04:49, 2.81it/s]
36%	449/1261 [02:37<04:42, 2.88it/s]
36%	450/1261 [02:37<04:49, 2.80it/s]
36%	451/1261 [02:37<04:43, 2.86it/s]
36%	452/1261 [02:38<04:45, 2.84it/s]
36%	453/1261 [02:38<04:46, 2.82it/s]
36%	454/1261 [02:38<04:39, 2.89it/s]
36%	455/1261 [02:39<04:45, 2.82it/s]
36%	456/1261 [02:39<04:38, 2.89it/s]
36%	457/1261 [02:39<04:46, 2.80it/s]
36%	458/1261 [02:40<04:40, 2.86it/s]
36%	459/1261 [02:40<04:46, 2.80it/s]
36%	460/1261 [02:41<04:40, 2.86it/s]
37%	461/1261 [02:41<04:46, 2.80it/s]
37%	462/1261 [02:41<04:38, 2.87it/s]

37%	463/1261 [02:42<04:48, 2.76it/s]
37%	464/1261 [02:42<04:42, 2.83it/s]
37%	465/1261 [02:42<04:47, 2.77it/s]
37%	466/1261 [02:43<04:39, 2.84it/s]
37%	467/1261 [02:43<04:44, 2.79it/s]
37%	468/1261 [02:43<04:38, 2.85it/s]
37%	469/1261 [02:44<04:46, 2.77it/s]
37%	470/1261 [02:44<04:38, 2.84it/s]
37%	471/1261 [02:44<04:43, 2.79it/s]
37%	472/1261 [02:45<04:34, 2.87it/s]
38%	473/1261 [02:45<04:41, 2.80it/s]
38%	474/1261 [02:46<04:36, 2.85it/s]
38%	475/1261 [02:46<04:40, 2.80it/s]
38%	476/1261 [02:46<04:34, 2.86it/s]
38%	477/1261 [02:47<04:39, 2.81it/s]
38%	478/1261 [02:47<04:33, 2.86it/s]
38%	479/1261 [02:47<04:38, 2.81it/s]
38%	480/1261 [02:48<04:33, 2.86it/s]
38%	481/1261 [02:48<04:39, 2.79it/s]
38%	482/1261 [02:48<04:32, 2.86it/s]
38%	483/1261 [02:49<04:38, 2.79it/s]
38%	484/1261 [02:49<04:31, 2.87it/s]
38%	485/1261 [02:49<04:37, 2.79it/s]
39%	486/1261 [02:50<04:31, 2.85it/s]
39%	487/1261 [02:50<04:38, 2.78it/s]
39%	488/1261 [02:50<04:33, 2.83it/s]
39%	489/1261 [02:51<04:27, 2.88it/s]
39%	490/1261 [02:51<04:37, 2.78it/s]
39%	491/1261 [02:52<04:32, 2.83it/s]
39%	492/1261 [02:52<04:27, 2.87it/s]
39%	493/1261 [02:52<04:38, 2.76it/s]
39%	494/1261 [02:53<04:31, 2.83it/s]
39%	495/1261 [02:53<04:35, 2.78it/s]
39%	496/1261 [02:53<04:28, 2.85it/s]
39%	497/1261 [02:54<04:32, 2.80it/s]
39%	498/1261 [02:54<04:25, 2.87it/s]
40%	499/1261 [02:54<04:32, 2.79it/s]
40%	500/1261 [02:55<04:28, 2.84it/s]
40%	501/1261 [02:55<04:24, 2.88it/s]
40%	502/1261 [02:55<04:31, 2.80it/s]
40%	503/1261 [02:56<04:24, 2.87it/s]
40%	504/1261 [02:56<04:30, 2.80it/s]
40%	505/1261 [02:56<04:24, 2.86it/s]
40%	506/1261 [02:57<04:28, 2.81it/s]
40%	507/1261 [02:57<04:22, 2.88it/s]
40%	508/1261 [02:58<04:27, 2.82it/s]
40%	509/1261 [02:58<04:19, 2.89it/s]
40%	510/1261 [02:58<04:30, 2.78it/s]

41%	511/1261 [02:59<04:22, 2.86it/s]
41%	512/1261 [02:59<04:28, 2.78it/s]
41%	513/1261 [02:59<04:24, 2.83it/s]
41%	514/1261 [03:00<04:29, 2.78it/s]
41%	515/1261 [03:00<04:22, 2.84it/s]
41%	516/1261 [03:00<04:28, 2.78it/s]
41%	517/1261 [03:01<04:23, 2.82it/s]
41%	518/1261 [03:01<04:29, 2.75it/s]
41%	519/1261 [03:01<04:22, 2.83it/s]
41%	520/1261 [03:02<04:27, 2.77it/s]
41%	521/1261 [03:02<04:21, 2.83it/s]
41%	522/1261 [03:03<04:30, 2.73it/s]
41%	523/1261 [03:03<04:23, 2.80it/s]
42%	524/1261 [03:03<04:29, 2.74it/s]
42%	525/1261 [03:04<04:22, 2.81it/s]
42%	526/1261 [03:04<04:28, 2.74it/s]
42%	527/1261 [03:04<04:22, 2.80it/s]
42%	528/1261 [03:05<04:26, 2.75it/s]
42%	529/1261 [03:05<04:21, 2.80it/s]
42%	530/1261 [03:05<04:26, 2.74it/s]
42%	531/1261 [03:06<04:20, 2.80it/s]
42%	532/1261 [03:06<04:22, 2.77it/s]
42%	533/1261 [03:07<04:25, 2.75it/s]
42%	534/1261 [03:07<04:24, 2.75it/s]
42%	535/1261 [03:07<04:26, 2.73it/s]
43%	536/1261 [03:08<04:33, 2.65it/s]
43%	537/1261 [03:08<04:25, 2.73it/s]
43%	538/1261 [03:08<04:26, 2.71it/s]
43%	539/1261 [03:09<04:26, 2.71it/s]
43%	540/1261 [03:09<04:26, 2.70it/s]
43%	541/1261 [03:09<04:25, 2.71it/s]
43%	542/1261 [03:10<04:23, 2.73it/s]
43%	543/1261 [03:10<04:20, 2.76it/s]
43%	544/1261 [03:11<04:16, 2.79it/s]
43%	545/1261 [03:11<04:10, 2.86it/s]
43%	546/1261 [03:11<04:14, 2.81it/s]
43%	547/1261 [03:12<04:18, 2.77it/s]
43%	548/1261 [03:12<04:21, 2.72it/s]
44%	549/1261 [03:12<04:24, 2.70it/s]
44%	550/1261 [03:13<04:23, 2.69it/s]
44%	551/1261 [03:13<04:28, 2.65it/s]
44%	552/1261 [03:13<04:22, 2.70it/s]
44%	553/1261 [03:14<04:33, 2.59it/s]
44%	554/1261 [03:14<04:28, 2.64it/s]
44%	555/1261 [03:15<04:30, 2.61it/s]
44%	556/1261 [03:15<04:35, 2.56it/s]
44%	557/1261 [03:15<04:28, 2.62it/s]
44%	558/1261 [03:16<04:36, 2.54it/s]

44%		559/1261	[03:16<04:29,	2.60it/s]
44%		560/1261	[03:17<04:36,	2.54it/s]
44%		561/1261	[03:17<04:29,	2.60it/s]
45%		562/1261	[03:17<04:34,	2.54it/s]
45%		563/1261	[03:18<04:26,	2.62it/s]
45%		564/1261	[03:18<04:30,	2.58it/s]
45%		565/1261	[03:19<04:23,	2.64it/s]
45%		566/1261	[03:19<04:29,	2.58it/s]
45%		567/1261	[03:19<04:23,	2.64it/s]
45%		568/1261	[03:20<04:27,	2.59it/s]
45%		569/1261	[03:20<04:22,	2.64it/s]
45%		570/1261	[03:20<04:28,	2.57it/s]
45%		571/1261	[03:21<04:23,	2.62it/s]
45%		572/1261	[03:21<04:27,	2.58it/s]
45%		573/1261	[03:22<04:20,	2.64it/s]
46%		574/1261	[03:22<04:24,	2.60it/s]
46%		575/1261	[03:22<04:18,	2.65it/s]
46%		576/1261	[03:23<04:22,	2.61it/s]
46%		577/1261	[03:23<04:16,	2.67it/s]
46%		578/1261	[03:24<04:22,	2.60it/s]
46%		579/1261	[03:24<04:18,	2.64it/s]
46%		580/1261	[03:24<04:23,	2.59it/s]
46%		581/1261	[03:25<04:16,	2.65it/s]
46%		582/1261	[03:25<04:17,	2.64it/s]
46%		583/1261	[03:25<04:27,	2.54it/s]
46%		584/1261	[03:26<04:20,	2.60it/s]
46%		585/1261	[03:26<04:23,	2.57it/s]
46%		586/1261	[03:27<04:14,	2.65it/s]
47%		587/1261	[03:27<04:17,	2.61it/s]
47%		588/1261	[03:27<04:11,	2.68it/s]
47%		589/1261	[03:28<04:10,	2.68it/s]
47%		590/1261	[03:28<04:10,	2.68it/s]
47%		591/1261	[03:28<04:07,	2.70it/s]
47%		592/1261	[03:29<04:06,	2.72it/s]
47%		593/1261	[03:29<04:06,	2.71it/s]
47%		594/1261	[03:30<04:07,	2.69it/s]
47%		595/1261	[03:30<04:07,	2.69it/s]
47%		596/1261	[03:30<04:08,	2.68it/s]
47%		597/1261	[03:31<04:09,	2.67it/s]
47%		598/1261	[03:31<04:08,	2.67it/s]
48%		599/1261	[03:31<04:08,	2.66it/s]
48%		600/1261	[03:32<04:13,	2.61it/s]
48%		601/1261	[03:32<04:10,	2.63it/s]
48%		602/1261	[03:33<04:09,	2.64it/s]
48%		603/1261	[03:33<04:07,	2.65it/s]
48%		604/1261	[03:33<04:05,	2.67it/s]
48%		605/1261	[03:34<04:07,	2.65it/s]
48%		606/1261	[03:34<04:08,	2.64it/s]

48%		607/1261	[03:34<04:12,	2.59it/s]
48%		608/1261	[03:35<04:14,	2.57it/s]
48%		609/1261	[03:35<04:06,	2.64it/s]
48%		610/1261	[03:36<04:10,	2.60it/s]
48%		611/1261	[03:36<04:02,	2.68it/s]
49%		612/1261	[03:36<04:03,	2.66it/s]
49%		613/1261	[03:37<04:11,	2.58it/s]
49%		614/1261	[03:37<04:06,	2.63it/s]
49%		615/1261	[03:38<04:11,	2.57it/s]
49%		616/1261	[03:38<04:09,	2.58it/s]
49%		617/1261	[03:38<04:01,	2.67it/s]
49%		618/1261	[03:39<04:03,	2.64it/s]
49%		619/1261	[03:39<04:04,	2.62it/s]
49%		620/1261	[03:39<03:58,	2.69it/s]
49%		621/1261	[03:40<04:01,	2.65it/s]
49%		622/1261	[03:40<03:56,	2.70it/s]
49%		623/1261	[03:41<03:57,	2.69it/s]
49%		624/1261	[03:41<03:57,	2.68it/s]
50%		625/1261	[03:41<04:00,	2.64it/s]
50%		626/1261	[03:42<03:53,	2.72it/s]
50%		627/1261	[03:42<03:58,	2.66it/s]
50%		628/1261	[03:42<03:51,	2.74it/s]
50%		629/1261	[03:43<03:56,	2.68it/s]
50%		630/1261	[03:43<03:51,	2.72it/s]
50%		631/1261	[03:44<03:57,	2.65it/s]
50%		632/1261	[03:44<03:50,	2.73it/s]
50%		633/1261	[03:44<03:57,	2.64it/s]
50%		634/1261	[03:45<03:50,	2.72it/s]
50%		635/1261	[03:45<03:53,	2.68it/s]
50%		636/1261	[03:45<03:46,	2.75it/s]
51%		637/1261	[03:46<03:51,	2.70it/s]
51%		638/1261	[03:46<03:45,	2.77it/s]
51%		639/1261	[03:46<03:49,	2.71it/s]
51%		640/1261	[03:47<03:43,	2.77it/s]
51%		641/1261	[03:47<03:47,	2.73it/s]
51%		642/1261	[03:48<03:54,	2.64it/s]
51%		643/1261	[03:48<03:46,	2.72it/s]
51%		644/1261	[03:48<03:49,	2.69it/s]
51%		645/1261	[03:49<03:42,	2.77it/s]
51%		646/1261	[03:49<03:47,	2.70it/s]
51%		647/1261	[03:49<03:46,	2.71it/s]
51%		648/1261	[03:50<03:49,	2.67it/s]
51%		649/1261	[03:50<03:50,	2.66it/s]
52%		650/1261	[03:50<03:43,	2.74it/s]
52%		651/1261	[03:51<03:42,	2.74it/s]
52%		652/1261	[03:51<03:42,	2.73it/s]
52%		653/1261	[03:52<03:41,	2.75it/s]
52%		654/1261	[03:52<03:43,	2.71it/s]

52%		655/1261	[03:52<03:38,	2.78it/s]
52%		656/1261	[03:53<03:39,	2.76it/s]
52%		657/1261	[03:53<03:38,	2.76it/s]
52%		658/1261	[03:53<03:37,	2.77it/s]
52%		659/1261	[03:54<03:36,	2.78it/s]
52%		660/1261	[03:54<03:37,	2.76it/s]
52%		661/1261	[03:54<03:39,	2.74it/s]
52%		662/1261	[03:55<03:40,	2.71it/s]
53%		663/1261	[03:55<03:39,	2.73it/s]
53%		664/1261	[03:56<03:40,	2.71it/s]
53%		665/1261	[03:56<03:38,	2.72it/s]
53%		666/1261	[03:56<03:38,	2.72it/s]
53%		667/1261	[03:57<03:38,	2.72it/s]
53%		668/1261	[03:57<03:38,	2.72it/s]
53%		669/1261	[03:57<03:37,	2.72it/s]
53%		670/1261	[03:58<03:37,	2.72it/s]
53%		671/1261	[03:58<03:51,	2.55it/s]
53%		672/1261	[03:59<03:43,	2.63it/s]
53%		673/1261	[03:59<03:47,	2.58it/s]
53%		674/1261	[03:59<03:40,	2.66it/s]
54%		675/1261	[04:00<03:42,	2.64it/s]
54%		676/1261	[04:00<03:34,	2.72it/s]
54%		677/1261	[04:00<03:39,	2.65it/s]
54%		678/1261	[04:01<03:34,	2.71it/s]
54%		679/1261	[04:01<03:39,	2.65it/s]
54%		680/1261	[04:02<03:33,	2.73it/s]
54%		681/1261	[04:02<03:37,	2.66it/s]
54%		682/1261	[04:02<03:32,	2.73it/s]
54%		683/1261	[04:03<03:35,	2.68it/s]
54%		684/1261	[04:03<03:29,	2.75it/s]
54%		685/1261	[04:03<03:31,	2.72it/s]
54%		686/1261	[04:04<03:34,	2.68it/s]
54%		687/1261	[04:04<03:28,	2.76it/s]
55%		688/1261	[04:05<03:27,	2.76it/s]
55%		689/1261	[04:05<03:25,	2.78it/s]
55%		690/1261	[04:05<03:26,	2.77it/s]
55%		691/1261	[04:06<03:28,	2.74it/s]
55%		692/1261	[04:06<03:25,	2.77it/s]
55%		693/1261	[04:06<03:29,	2.71it/s]
55%		694/1261	[04:07<03:23,	2.78it/s]
55%		695/1261	[04:07<03:27,	2.73it/s]
55%		696/1261	[04:07<03:21,	2.81it/s]
55%		697/1261	[04:08<03:24,	2.76it/s]
55%		698/1261	[04:08<03:18,	2.84it/s]
55%		699/1261	[04:08<03:23,	2.77it/s]
56%		700/1261	[04:09<03:20,	2.80it/s]
56%		701/1261	[04:09<03:26,	2.71it/s]
56%		702/1261	[04:10<03:22,	2.77it/s]

56%	703/1261 [04:10<03:32,	2.63it/s]
56%	704/1261 [04:10<03:26,	2.70it/s]
56%	705/1261 [04:11<03:31,	2.62it/s]
56%	706/1261 [04:11<03:27,	2.68it/s]
56%	707/1261 [04:11<03:27,	2.67it/s]
56%	708/1261 [04:12<03:20,	2.75it/s]
56%	709/1261 [04:12<03:22,	2.72it/s]
56%	710/1261 [04:13<03:15,	2.81it/s]
56%	711/1261 [04:13<03:18,	2.77it/s]
56%	712/1261 [04:13<03:11,	2.86it/s]
57%	713/1261 [04:14<03:14,	2.82it/s]
57%	714/1261 [04:14<03:10,	2.87it/s]
57%	715/1261 [04:14<03:13,	2.82it/s]
57%	716/1261 [04:15<03:08,	2.90it/s]
57%	717/1261 [04:15<03:11,	2.84it/s]
57%	718/1261 [04:15<03:07,	2.90it/s]
57%	719/1261 [04:16<03:09,	2.85it/s]
57%	720/1261 [04:16<03:06,	2.90it/s]
57%	721/1261 [04:16<03:11,	2.82it/s]
57%	722/1261 [04:17<03:06,	2.89it/s]
57%	723/1261 [04:17<03:10,	2.82it/s]
57%	724/1261 [04:17<03:04,	2.91it/s]
57%	725/1261 [04:18<03:08,	2.84it/s]
58%	726/1261 [04:18<03:05,	2.89it/s]
58%	727/1261 [04:18<03:02,	2.92it/s]
58%	728/1261 [04:19<03:08,	2.82it/s]
58%	729/1261 [04:19<03:06,	2.86it/s]
58%	730/1261 [04:19<03:04,	2.88it/s]
58%	731/1261 [04:20<03:06,	2.84it/s]
58%	732/1261 [04:20<03:02,	2.90it/s]
58%	733/1261 [04:21<03:08,	2.79it/s]
58%	734/1261 [04:21<03:04,	2.86it/s]
58%	735/1261 [04:21<03:06,	2.81it/s]
58%	736/1261 [04:22<03:01,	2.89it/s]
58%	737/1261 [04:22<03:06,	2.82it/s]
59%	738/1261 [04:22<03:02,	2.86it/s]
59%	739/1261 [04:23<02:59,	2.91it/s]
59%	740/1261 [04:23<03:04,	2.83it/s]
59%	741/1261 [04:23<02:59,	2.89it/s]
59%	742/1261 [04:24<03:04,	2.82it/s]
59%	743/1261 [04:24<03:01,	2.85it/s]
59%	744/1261 [04:24<03:00,	2.87it/s]
59%	745/1261 [04:25<03:05,	2.78it/s]
59%	746/1261 [04:25<03:03,	2.81it/s]
59%	747/1261 [04:26<03:07,	2.75it/s]
59%	748/1261 [04:26<03:03,	2.80it/s]
59%	749/1261 [04:26<03:07,	2.73it/s]
59%	750/1261 [04:27<03:03,	2.79it/s]

60%		751/1261	[04:27<03:09,	2.70it/s]
60%		752/1261	[04:27<03:05,	2.75it/s]
60%		753/1261	[04:28<03:09,	2.68it/s]
60%		754/1261	[04:28<03:04,	2.75it/s]
60%		755/1261	[04:28<03:08,	2.68it/s]
60%		756/1261	[04:29<03:05,	2.72it/s]
60%		757/1261	[04:29<03:09,	2.66it/s]
60%		758/1261	[04:30<03:03,	2.74it/s]
60%		759/1261	[04:30<03:05,	2.71it/s]
60%		760/1261	[04:30<03:00,	2.77it/s]
60%		761/1261	[04:31<03:06,	2.68it/s]
60%		762/1261	[04:31<03:03,	2.72it/s]
61%		763/1261	[04:31<03:06,	2.67it/s]
61%		764/1261	[04:32<03:01,	2.74it/s]
61%		765/1261	[04:32<03:04,	2.69it/s]
61%		766/1261	[04:32<02:58,	2.77it/s]
61%		767/1261	[04:33<03:00,	2.74it/s]
61%		768/1261	[04:33<02:57,	2.78it/s]
61%		769/1261	[04:34<03:03,	2.69it/s]
61%		770/1261	[04:34<02:59,	2.74it/s]
61%		771/1261	[04:34<03:03,	2.67it/s]
61%		772/1261	[04:35<03:01,	2.69it/s]
61%		773/1261	[04:35<02:57,	2.75it/s]
61%		774/1261	[04:35<03:01,	2.69it/s]
61%		775/1261	[04:36<02:55,	2.77it/s]
62%		776/1261	[04:36<02:59,	2.70it/s]
62%		777/1261	[04:37<02:57,	2.73it/s]
62%		778/1261	[04:37<02:55,	2.76it/s]
62%		779/1261	[04:37<03:00,	2.67it/s]
62%		780/1261	[04:38<02:56,	2.73it/s]
62%		781/1261	[04:38<03:00,	2.67it/s]
62%		782/1261	[04:38<02:54,	2.74it/s]
62%		783/1261	[04:39<02:57,	2.70it/s]
62%		784/1261	[04:39<02:52,	2.77it/s]
62%		785/1261	[04:40<02:56,	2.70it/s]
62%		786/1261	[04:40<02:50,	2.79it/s]
62%		787/1261	[04:40<02:53,	2.73it/s]
62%		788/1261	[04:41<02:49,	2.79it/s]
63%		789/1261	[04:41<02:52,	2.73it/s]
63%		790/1261	[04:41<02:48,	2.79it/s]
63%		791/1261	[04:42<02:51,	2.74it/s]
63%		792/1261	[04:42<02:48,	2.79it/s]
63%		793/1261	[04:42<02:58,	2.62it/s]
63%		794/1261	[04:43<02:52,	2.71it/s]
63%		795/1261	[04:43<02:46,	2.80it/s]
63%		796/1261	[04:43<02:48,	2.77it/s]
63%		797/1261	[04:44<02:44,	2.82it/s]
63%		798/1261	[04:44<02:47,	2.76it/s]

63%		799/1261	[04:45<02:43,	2.82it/s]
63%		800/1261	[04:45<02:46,	2.76it/s]
64%		801/1261	[04:45<02:43,	2.81it/s]
64%		802/1261	[04:46<02:48,	2.72it/s]
64%		803/1261	[04:46<02:47,	2.74it/s]
64%		804/1261	[04:46<02:45,	2.77it/s]
64%		805/1261	[04:47<02:47,	2.72it/s]
64%		806/1261	[04:47<02:42,	2.79it/s]
64%		807/1261	[04:47<02:46,	2.73it/s]
64%		808/1261	[04:48<02:40,	2.82it/s]
64%		809/1261	[04:48<02:42,	2.78it/s]
64%		810/1261	[04:49<02:39,	2.83it/s]
64%		811/1261	[04:49<02:42,	2.77it/s]
64%		812/1261	[04:49<02:38,	2.84it/s]
64%		813/1261	[04:50<02:41,	2.77it/s]
65%		814/1261	[04:50<02:37,	2.84it/s]
65%		815/1261	[04:50<02:40,	2.77it/s]
65%		816/1261	[04:51<02:38,	2.81it/s]
65%		817/1261	[04:51<02:35,	2.86it/s]
65%		818/1261	[04:51<02:38,	2.79it/s]
65%		819/1261	[04:52<02:35,	2.85it/s]
65%		820/1261	[04:52<02:36,	2.81it/s]
65%		821/1261	[04:52<02:32,	2.88it/s]
65%		822/1261	[04:53<02:37,	2.78it/s]
65%		823/1261	[04:53<02:33,	2.85it/s]
65%		824/1261	[04:53<02:36,	2.79it/s]
65%		825/1261	[04:54<02:32,	2.85it/s]
66%		826/1261	[04:54<02:36,	2.78it/s]
66%		827/1261	[04:55<02:35,	2.80it/s]
66%		828/1261	[04:55<02:37,	2.75it/s]
66%		829/1261	[04:55<02:33,	2.82it/s]
66%		830/1261	[04:56<02:36,	2.75it/s]
66%		831/1261	[04:56<02:34,	2.79it/s]
66%		832/1261	[04:56<02:31,	2.83it/s]
66%		833/1261	[04:57<02:34,	2.78it/s]
66%		834/1261	[04:57<02:30,	2.83it/s]
66%		835/1261	[04:57<02:33,	2.78it/s]
66%		836/1261	[04:58<02:29,	2.84it/s]
66%		837/1261	[04:58<02:32,	2.77it/s]
66%		838/1261	[04:58<02:28,	2.84it/s]
67%		839/1261	[04:59<02:32,	2.76it/s]
67%		840/1261	[04:59<02:31,	2.78it/s]
67%		841/1261	[05:00<02:29,	2.81it/s]
67%		842/1261	[05:00<02:32,	2.74it/s]
67%		843/1261	[05:00<02:28,	2.81it/s]
67%		844/1261	[05:01<02:30,	2.76it/s]
67%		845/1261	[05:01<02:26,	2.84it/s]
67%		846/1261	[05:01<02:29,	2.78it/s]

67%	847/1261 [05:02<02:24,	2.86it/s]
67%	848/1261 [05:02<02:27,	2.81it/s]
67%	849/1261 [05:02<02:23,	2.86it/s]
67%	850/1261 [05:03<02:26,	2.81it/s]
67%	851/1261 [05:03<02:24,	2.83it/s]
68%	852/1261 [05:03<02:22,	2.86it/s]
68%	853/1261 [05:04<02:28,	2.75it/s]
68%	854/1261 [05:04<02:25,	2.81it/s]
68%	855/1261 [05:05<02:26,	2.78it/s]
68%	856/1261 [05:05<02:21,	2.86it/s]
68%	857/1261 [05:05<02:25,	2.78it/s]
68%	858/1261 [05:06<02:21,	2.85it/s]
68%	859/1261 [05:06<02:24,	2.79it/s]
68%	860/1261 [05:06<02:20,	2.86it/s]
68%	861/1261 [05:07<02:22,	2.81it/s]
68%	862/1261 [05:07<02:18,	2.89it/s]
68%	863/1261 [05:07<02:20,	2.82it/s]
69%	864/1261 [05:08<02:17,	2.89it/s]
69%	865/1261 [05:08<02:20,	2.82it/s]
69%	866/1261 [05:08<02:17,	2.88it/s]
69%	867/1261 [05:09<02:19,	2.82it/s]
69%	868/1261 [05:09<02:16,	2.88it/s]
69%	869/1261 [05:09<02:19,	2.81it/s]
69%	870/1261 [05:10<02:16,	2.86it/s]
69%	871/1261 [05:10<02:19,	2.79it/s]
69%	872/1261 [05:11<02:16,	2.85it/s]
69%	873/1261 [05:11<02:19,	2.78it/s]
69%	874/1261 [05:11<02:15,	2.85it/s]
69%	875/1261 [05:12<02:17,	2.81it/s]
69%	876/1261 [05:12<02:14,	2.87it/s]
70%	877/1261 [05:12<02:16,	2.81it/s]
70%	878/1261 [05:13<02:13,	2.87it/s]
70%	879/1261 [05:13<02:15,	2.82it/s]
70%	880/1261 [05:13<02:12,	2.88it/s]
70%	881/1261 [05:14<02:14,	2.82it/s]
70%	882/1261 [05:14<02:11,	2.87it/s]
70%	883/1261 [05:14<02:16,	2.78it/s]
70%	884/1261 [05:15<02:13,	2.83it/s]
70%	885/1261 [05:15<02:15,	2.78it/s]
70%	886/1261 [05:15<02:11,	2.86it/s]
70%	887/1261 [05:16<02:13,	2.80it/s]
70%	888/1261 [05:16<02:09,	2.87it/s]
70%	889/1261 [05:17<02:12,	2.81it/s]
71%	890/1261 [05:17<02:08,	2.88it/s]
71%	891/1261 [05:17<02:10,	2.83it/s]
71%	892/1261 [05:18<02:07,	2.89it/s]
71%	893/1261 [05:18<02:09,	2.85it/s]
71%	894/1261 [05:18<02:06,	2.90it/s]

71%	895/1261 [05:19<02:09,	2.82it/s]
71%	896/1261 [05:19<02:06,	2.88it/s]
71%	897/1261 [05:19<02:10,	2.80it/s]
71%	898/1261 [05:20<02:06,	2.86it/s]
71%	899/1261 [05:20<02:09,	2.79it/s]
71%	900/1261 [05:20<02:08,	2.81it/s]
71%	901/1261 [05:21<02:11,	2.75it/s]
72%	902/1261 [05:21<02:07,	2.81it/s]
72%	903/1261 [05:22<02:09,	2.76it/s]
72%	904/1261 [05:22<02:05,	2.85it/s]
72%	905/1261 [05:22<02:07,	2.78it/s]
72%	906/1261 [05:23<02:04,	2.85it/s]
72%	907/1261 [05:23<02:06,	2.81it/s]
72%	908/1261 [05:23<02:03,	2.86it/s]
72%	909/1261 [05:24<02:05,	2.79it/s]
72%	910/1261 [05:24<02:02,	2.86it/s]
72%	911/1261 [05:24<02:03,	2.83it/s]
72%	912/1261 [05:25<02:06,	2.76it/s]
72%	913/1261 [05:25<02:03,	2.82it/s]
72%	914/1261 [05:25<02:05,	2.77it/s]
73%	915/1261 [05:26<02:01,	2.85it/s]
73%	916/1261 [05:26<02:02,	2.81it/s]
73%	917/1261 [05:26<01:59,	2.88it/s]
73%	918/1261 [05:27<02:00,	2.84it/s]
73%	919/1261 [05:27<01:58,	2.89it/s]
73%	920/1261 [05:27<01:59,	2.85it/s]
73%	921/1261 [05:28<02:00,	2.81it/s]
73%	922/1261 [05:28<01:58,	2.87it/s]
73%	923/1261 [05:29<02:00,	2.80it/s]
73%	924/1261 [05:29<01:57,	2.86it/s]
73%	925/1261 [05:29<02:00,	2.78it/s]
73%	926/1261 [05:30<01:57,	2.86it/s]
74%	927/1261 [05:30<01:59,	2.78it/s]
74%	928/1261 [05:30<01:57,	2.84it/s]
74%	929/1261 [05:31<01:59,	2.78it/s]
74%	930/1261 [05:31<01:57,	2.83it/s]
74%	931/1261 [05:31<01:59,	2.77it/s]
74%	932/1261 [05:32<01:56,	2.84it/s]
74%	933/1261 [05:32<01:58,	2.77it/s]
74%	934/1261 [05:32<01:54,	2.85it/s]
74%	935/1261 [05:33<01:56,	2.81it/s]
74%	936/1261 [05:33<01:52,	2.88it/s]
74%	937/1261 [05:34<01:55,	2.80it/s]
74%	938/1261 [05:34<01:52,	2.87it/s]
74%	939/1261 [05:34<01:54,	2.80it/s]
75%	940/1261 [05:35<01:51,	2.87it/s]
75%	941/1261 [05:35<01:54,	2.80it/s]
75%	942/1261 [05:35<01:51,	2.86it/s]

75%		943/1261	[05:36<01:54,	2.78it/s]
75%		944/1261	[05:36<01:51,	2.84it/s]
75%		945/1261	[05:36<01:54,	2.76it/s]
75%		946/1261	[05:37<01:51,	2.81it/s]
75%		947/1261	[05:37<01:53,	2.76it/s]
75%		948/1261	[05:37<01:50,	2.83it/s]
75%		949/1261	[05:38<01:52,	2.78it/s]
75%		950/1261	[05:38<01:49,	2.83it/s]
75%		951/1261	[05:39<01:51,	2.79it/s]
75%		952/1261	[05:39<01:48,	2.85it/s]
76%		953/1261	[05:39<01:51,	2.76it/s]
76%		954/1261	[05:40<01:48,	2.82it/s]
76%		955/1261	[05:40<01:51,	2.75it/s]
76%		956/1261	[05:40<01:48,	2.82it/s]
76%		957/1261	[05:41<01:51,	2.73it/s]
76%		958/1261	[05:41<01:48,	2.79it/s]
76%		959/1261	[05:41<01:50,	2.74it/s]
76%		960/1261	[05:42<01:48,	2.78it/s]
76%		961/1261	[05:42<01:50,	2.71it/s]
76%		962/1261	[05:42<01:47,	2.77it/s]
76%		963/1261	[05:43<01:50,	2.70it/s]
76%		964/1261	[05:43<01:47,	2.77it/s]
77%		965/1261	[05:44<01:49,	2.70it/s]
77%		966/1261	[05:44<01:46,	2.76it/s]
77%		967/1261	[05:44<01:49,	2.69it/s]
77%		968/1261	[05:45<01:46,	2.74it/s]
77%		969/1261	[05:45<01:48,	2.68it/s]
77%		970/1261	[05:45<01:46,	2.74it/s]
77%		971/1261	[05:46<01:47,	2.69it/s]
77%		972/1261	[05:46<01:45,	2.74it/s]
77%		973/1261	[05:47<01:48,	2.66it/s]
77%		974/1261	[05:47<01:45,	2.72it/s]
77%		975/1261	[05:47<01:47,	2.67it/s]
77%		976/1261	[05:48<01:44,	2.72it/s]
77%		977/1261	[05:48<01:46,	2.66it/s]
78%		978/1261	[05:48<01:43,	2.72it/s]
78%		979/1261	[05:49<01:45,	2.68it/s]
78%		980/1261	[05:49<01:42,	2.75it/s]
78%		981/1261	[05:50<01:44,	2.68it/s]
78%		982/1261	[05:50<01:41,	2.74it/s]
78%		983/1261	[05:50<01:43,	2.68it/s]
78%		984/1261	[05:51<01:41,	2.74it/s]
78%		985/1261	[05:51<01:43,	2.68it/s]
78%		986/1261	[05:51<01:40,	2.73it/s]
78%		987/1261	[05:52<01:42,	2.67it/s]
78%		988/1261	[05:52<01:39,	2.74it/s]
78%		989/1261	[05:52<01:41,	2.67it/s]
79%		990/1261	[05:53<01:40,	2.71it/s]

79%		991/1261	[05:53<01:42,	2.65it/s]
79%		992/1261	[05:54<01:39,	2.70it/s]
79%		993/1261	[05:54<01:40,	2.68it/s]
79%		994/1261	[05:54<01:40,	2.66it/s]
79%		995/1261	[05:55<01:39,	2.67it/s]
79%		996/1261	[05:55<01:41,	2.62it/s]
79%		997/1261	[05:55<01:38,	2.68it/s]
79%		998/1261	[05:56<01:38,	2.66it/s]
79%		999/1261	[05:56<01:38,	2.65it/s]
79%		1000/1261	[05:57<01:39,	2.62it/s]
79%		1001/1261	[05:57<01:39,	2.62it/s]
79%		1002/1261	[05:57<01:38,	2.63it/s]
80%		1003/1261	[05:58<01:40,	2.56it/s]
80%		1004/1261	[05:58<01:38,	2.62it/s]
80%		1005/1261	[05:59<01:38,	2.60it/s]
80%		1006/1261	[05:59<01:38,	2.59it/s]
80%		1007/1261	[05:59<01:39,	2.55it/s]
80%		1008/1261	[06:00<01:37,	2.60it/s]
80%		1009/1261	[06:00<01:39,	2.52it/s]
80%		1010/1261	[06:01<01:37,	2.58it/s]
80%		1011/1261	[06:01<01:39,	2.51it/s]
80%		1012/1261	[06:01<01:37,	2.56it/s]
80%		1013/1261	[06:02<01:39,	2.50it/s]
80%		1014/1261	[06:02<01:37,	2.54it/s]
80%		1015/1261	[06:03<01:39,	2.46it/s]
81%		1016/1261	[06:03<01:37,	2.51it/s]
81%		1017/1261	[06:03<01:38,	2.47it/s]
81%		1018/1261	[06:04<01:36,	2.52it/s]
81%		1019/1261	[06:04<01:37,	2.47it/s]
81%		1020/1261	[06:05<01:35,	2.52it/s]
81%		1021/1261	[06:05<01:36,	2.49it/s]
81%		1022/1261	[06:05<01:34,	2.54it/s]
81%		1023/1261	[06:06<01:33,	2.53it/s]
81%		1024/1261	[06:06<01:35,	2.49it/s]
81%		1025/1261	[06:07<01:32,	2.54it/s]
81%		1026/1261	[06:07<01:32,	2.54it/s]
81%		1027/1261	[06:07<01:31,	2.57it/s]
82%		1028/1261	[06:08<01:29,	2.60it/s]
82%		1029/1261	[06:08<01:29,	2.58it/s]
82%		1030/1261	[06:08<01:29,	2.59it/s]
82%		1031/1261	[06:09<01:29,	2.58it/s]
82%		1032/1261	[06:09<01:26,	2.63it/s]
82%		1033/1261	[06:10<01:29,	2.56it/s]
82%		1034/1261	[06:10<01:28,	2.56it/s]
82%		1035/1261	[06:10<01:25,	2.64it/s]
82%		1036/1261	[06:11<01:25,	2.63it/s]
82%		1037/1261	[06:11<01:25,	2.62it/s]
82%		1038/1261	[06:11<01:24,	2.64it/s]

82%		1039/1261	[06:12<01:24,	2.63it/s]
82%		1040/1261	[06:12<01:23,	2.65it/s]
83%		1041/1261	[06:13<01:23,	2.64it/s]
83%		1042/1261	[06:13<01:22,	2.65it/s]
83%		1043/1261	[06:13<01:20,	2.72it/s]
83%		1044/1261	[06:14<01:18,	2.76it/s]
83%		1045/1261	[06:14<01:16,	2.84it/s]
83%		1046/1261	[06:14<01:16,	2.81it/s]
83%		1047/1261	[06:15<01:16,	2.78it/s]
83%		1048/1261	[06:15<01:17,	2.74it/s]
83%		1049/1261	[06:16<01:17,	2.73it/s]
83%		1050/1261	[06:16<01:18,	2.70it/s]
83%		1051/1261	[06:16<01:18,	2.69it/s]
83%		1052/1261	[06:17<01:17,	2.69it/s]
84%		1053/1261	[06:17<01:17,	2.69it/s]
84%		1054/1261	[06:17<01:16,	2.70it/s]
84%		1055/1261	[06:18<01:17,	2.65it/s]
84%		1056/1261	[06:18<01:17,	2.65it/s]
84%		1057/1261	[06:19<01:16,	2.65it/s]
84%		1058/1261	[06:19<01:17,	2.64it/s]
84%		1059/1261	[06:19<01:17,	2.62it/s]
84%		1060/1261	[06:20<01:16,	2.64it/s]
84%		1061/1261	[06:20<01:16,	2.61it/s]
84%		1062/1261	[06:20<01:16,	2.60it/s]
84%		1063/1261	[06:21<01:21,	2.43it/s]
84%		1064/1261	[06:21<01:19,	2.49it/s]
84%		1065/1261	[06:22<01:18,	2.51it/s]
85%		1066/1261	[06:22<01:18,	2.48it/s]
85%		1067/1261	[06:22<01:16,	2.54it/s]
85%		1068/1261	[06:23<01:17,	2.49it/s]
85%		1069/1261	[06:23<01:15,	2.54it/s]
85%		1070/1261	[06:24<01:16,	2.49it/s]
85%		1071/1261	[06:24<01:14,	2.54it/s]
85%		1072/1261	[06:24<01:16,	2.48it/s]
85%		1073/1261	[06:25<01:14,	2.53it/s]
85%		1074/1261	[06:25<01:15,	2.47it/s]
85%		1075/1261	[06:26<01:13,	2.52it/s]
85%		1076/1261	[06:26<01:14,	2.47it/s]
85%		1077/1261	[06:26<01:12,	2.53it/s]
85%		1078/1261	[06:27<01:13,	2.48it/s]
86%		1079/1261	[06:27<01:11,	2.53it/s]
86%		1080/1261	[06:28<01:13,	2.47it/s]
86%		1081/1261	[06:28<01:11,	2.53it/s]
86%		1082/1261	[06:28<01:10,	2.54it/s]
86%		1083/1261	[06:29<01:10,	2.54it/s]
86%		1084/1261	[06:29<01:09,	2.53it/s]
86%		1085/1261	[06:30<01:08,	2.55it/s]
86%		1086/1261	[06:30<01:09,	2.54it/s]

86%		1087/1261	[06:30<01:07,	2.60it/s]
86%		1088/1261	[06:31<01:06,	2.60it/s]
86%		1089/1261	[06:31<01:05,	2.62it/s]
86%		1090/1261	[06:32<01:04,	2.63it/s]
87%		1091/1261	[06:32<01:03,	2.68it/s]
87%		1092/1261	[06:32<01:02,	2.69it/s]
87%		1093/1261	[06:33<01:04,	2.62it/s]
87%		1094/1261	[06:33<01:02,	2.66it/s]
87%		1095/1261	[06:33<01:02,	2.68it/s]
87%		1096/1261	[06:34<01:00,	2.71it/s]
87%		1097/1261	[06:34<00:59,	2.74it/s]
87%		1098/1261	[06:34<00:59,	2.76it/s]
87%		1099/1261	[06:35<00:58,	2.75it/s]
87%		1100/1261	[06:35<00:59,	2.73it/s]
87%		1101/1261	[06:36<00:58,	2.74it/s]
87%		1102/1261	[06:36<00:58,	2.72it/s]
87%		1103/1261	[06:36<00:56,	2.80it/s]
88%		1104/1261	[06:37<00:56,	2.79it/s]
88%		1105/1261	[06:37<00:55,	2.81it/s]
88%		1106/1261	[06:37<00:56,	2.76it/s]
88%		1107/1261	[06:38<00:54,	2.84it/s]
88%		1108/1261	[06:38<00:54,	2.78it/s]
88%		1109/1261	[06:38<00:53,	2.86it/s]
88%		1110/1261	[06:39<00:54,	2.78it/s]
88%		1111/1261	[06:39<00:52,	2.83it/s]
88%		1112/1261	[06:39<00:53,	2.79it/s]
88%		1113/1261	[06:40<00:51,	2.85it/s]
88%		1114/1261	[06:40<00:52,	2.79it/s]
88%		1115/1261	[06:41<00:51,	2.84it/s]
89%		1116/1261	[06:41<00:51,	2.80it/s]
89%		1117/1261	[06:41<00:50,	2.85it/s]
89%		1118/1261	[06:42<00:51,	2.78it/s]
89%		1119/1261	[06:42<00:50,	2.83it/s]
89%		1120/1261	[06:42<00:50,	2.80it/s]
89%		1121/1261	[06:43<00:50,	2.79it/s]
89%		1122/1261	[06:43<00:48,	2.85it/s]
89%		1123/1261	[06:43<00:50,	2.73it/s]
89%		1124/1261	[06:44<00:52,	2.60it/s]
89%		1125/1261	[06:44<00:50,	2.72it/s]
89%		1126/1261	[06:45<00:49,	2.72it/s]
89%		1127/1261	[06:45<00:47,	2.81it/s]
89%		1128/1261	[06:45<00:48,	2.76it/s]
90%		1129/1261	[06:46<00:46,	2.83it/s]
90%		1130/1261	[06:46<00:46,	2.80it/s]
90%		1131/1261	[06:46<00:45,	2.86it/s]
90%		1132/1261	[06:47<00:46,	2.80it/s]
90%		1133/1261	[06:47<00:44,	2.87it/s]
90%		1134/1261	[06:47<00:45,	2.80it/s]

90% | 1135/1261 [06:48<00:43, 2.87it/s]
90% | 1136/1261 [06:48<00:44, 2.80it/s]
90% | 1137/1261 [06:48<00:43, 2.86it/s]
90% | 1138/1261 [06:49<00:43, 2.82it/s]
90% | 1139/1261 [06:49<00:42, 2.87it/s]
90% | 1140/1261 [06:49<00:43, 2.79it/s]
90% | 1141/1261 [06:50<00:42, 2.84it/s]
91% | 1142/1261 [06:50<00:42, 2.78it/s]
91% | 1143/1261 [06:51<00:41, 2.84it/s]
91% | 1144/1261 [06:51<00:42, 2.78it/s]
91% | 1145/1261 [06:51<00:40, 2.83it/s]
91% | 1146/1261 [06:52<00:41, 2.78it/s]
91% | 1147/1261 [06:52<00:39, 2.85it/s]
91% | 1148/1261 [06:52<00:40, 2.80it/s]
91% | 1149/1261 [06:53<00:39, 2.85it/s]
91% | 1150/1261 [06:53<00:39, 2.80it/s]
91% | 1151/1261 [06:53<00:38, 2.85it/s]
91% | 1152/1261 [06:54<00:38, 2.84it/s]
91% | 1153/1261 [06:54<00:39, 2.74it/s]
92% | 1154/1261 [06:54<00:38, 2.80it/s]
92% | 1155/1261 [06:55<00:38, 2.78it/s]
92% | 1156/1261 [06:55<00:37, 2.83it/s]
92% | 1157/1261 [06:56<00:36, 2.83it/s]
92% | 1158/1261 [06:56<00:37, 2.77it/s]
92% | 1159/1261 [06:56<00:36, 2.83it/s]
92% | 1160/1261 [06:57<00:36, 2.79it/s]
92% | 1161/1261 [06:57<00:34, 2.86it/s]
92% | 1162/1261 [06:57<00:35, 2.81it/s]
92% | 1163/1261 [06:58<00:34, 2.85it/s]
92% | 1164/1261 [06:58<00:35, 2.77it/s]
92% | 1165/1261 [06:58<00:33, 2.83it/s]
92% | 1166/1261 [06:59<00:34, 2.77it/s]
93% | 1167/1261 [06:59<00:33, 2.84it/s]
93% | 1168/1261 [06:59<00:33, 2.76it/s]
93% | 1169/1261 [07:00<00:32, 2.83it/s]
93% | 1170/1261 [07:00<00:33, 2.75it/s]
93% | 1171/1261 [07:01<00:31, 2.81it/s]
93% | 1172/1261 [07:01<00:32, 2.75it/s]
93% | 1173/1261 [07:01<00:31, 2.82it/s]
93% | 1174/1261 [07:02<00:31, 2.76it/s]
93% | 1175/1261 [07:02<00:30, 2.82it/s]
93% | 1176/1261 [07:02<00:30, 2.77it/s]
93% | 1177/1261 [07:03<00:29, 2.85it/s]
93% | 1178/1261 [07:03<00:29, 2.78it/s]
93% | 1179/1261 [07:03<00:28, 2.84it/s]
94% | 1180/1261 [07:04<00:29, 2.79it/s]
94% | 1181/1261 [07:04<00:28, 2.85it/s]
94% | 1182/1261 [07:04<00:28, 2.74it/s]

94%|| 1183/1261 [07:05<00:27, 2.80it/s]
94%|| 1184/1261 [07:05<00:27, 2.76it/s]
94%|| 1185/1261 [07:06<00:26, 2.82it/s]
94%|| 1186/1261 [07:06<00:27, 2.76it/s]
94%|| 1187/1261 [07:06<00:26, 2.83it/s]
94%|| 1188/1261 [07:07<00:26, 2.78it/s]
94%|| 1189/1261 [07:07<00:25, 2.86it/s]
94%|| 1190/1261 [07:07<00:25, 2.80it/s]
94%|| 1191/1261 [07:08<00:24, 2.86it/s]
95%|| 1192/1261 [07:08<00:24, 2.80it/s]
95%|| 1193/1261 [07:08<00:23, 2.85it/s]
95%|| 1194/1261 [07:09<00:24, 2.79it/s]
95%|| 1195/1261 [07:09<00:23, 2.85it/s]
95%|| 1196/1261 [07:09<00:23, 2.78it/s]
95%|| 1197/1261 [07:10<00:22, 2.84it/s]
95%|| 1198/1261 [07:10<00:22, 2.77it/s]
95%|| 1199/1261 [07:10<00:21, 2.83it/s]
95%|| 1200/1261 [07:11<00:22, 2.74it/s]
95%|| 1201/1261 [07:11<00:21, 2.80it/s]
95%|| 1202/1261 [07:12<00:21, 2.75it/s]
95%|| 1203/1261 [07:12<00:20, 2.83it/s]
95%|| 1204/1261 [07:12<00:20, 2.77it/s]
96%|| 1205/1261 [07:13<00:19, 2.84it/s]
96%|| 1206/1261 [07:13<00:19, 2.78it/s]
96%|| 1207/1261 [07:13<00:18, 2.85it/s]
96%|| 1208/1261 [07:14<00:18, 2.79it/s]
96%|| 1209/1261 [07:14<00:18, 2.83it/s]
96%|| 1210/1261 [07:14<00:18, 2.78it/s]
96%|| 1211/1261 [07:15<00:17, 2.84it/s]
96%|| 1212/1261 [07:15<00:17, 2.82it/s]
96%|| 1213/1261 [07:16<00:17, 2.74it/s]
96%|| 1214/1261 [07:16<00:17, 2.73it/s]
96%|| 1215/1261 [07:16<00:16, 2.81it/s]
96%|| 1216/1261 [07:17<00:16, 2.79it/s]
97%|| 1217/1261 [07:17<00:15, 2.77it/s]
97%|| 1218/1261 [07:17<00:15, 2.85it/s]
97%|| 1219/1261 [07:18<00:14, 2.81it/s]
97%|| 1220/1261 [07:18<00:14, 2.87it/s]
97%|| 1221/1261 [07:18<00:14, 2.84it/s]
97%|| 1222/1261 [07:19<00:13, 2.80it/s]
97%|| 1223/1261 [07:19<00:13, 2.86it/s]
97%|| 1224/1261 [07:19<00:13, 2.79it/s]
97%|| 1225/1261 [07:20<00:12, 2.85it/s]
97%|| 1226/1261 [07:20<00:12, 2.83it/s]
97%|| 1227/1261 [07:20<00:12, 2.81it/s]
97%|| 1228/1261 [07:21<00:11, 2.79it/s]
97%|| 1229/1261 [07:21<00:11, 2.86it/s]
98%|| 1230/1261 [07:22<00:11, 2.78it/s]

```
98%|| 1231/1261 [07:22<00:10, 2.83it/s]
98%|| 1232/1261 [07:22<00:10, 2.78it/s]
98%|| 1233/1261 [07:23<00:09, 2.84it/s]
98%|| 1234/1261 [07:23<00:09, 2.77it/s]
98%|| 1235/1261 [07:23<00:09, 2.85it/s]
98%|| 1236/1261 [07:24<00:08, 2.78it/s]
98%|| 1237/1261 [07:24<00:08, 2.84it/s]
98%|| 1238/1261 [07:24<00:08, 2.80it/s]
98%|| 1239/1261 [07:25<00:07, 2.86it/s]
98%|| 1240/1261 [07:25<00:07, 2.78it/s]
98%|| 1241/1261 [07:25<00:07, 2.84it/s]
98%|| 1242/1261 [07:26<00:06, 2.72it/s]
99%|| 1243/1261 [07:26<00:06, 2.79it/s]
99%|| 1244/1261 [07:27<00:06, 2.74it/s]
99%|| 1245/1261 [07:27<00:05, 2.81it/s]
99%|| 1246/1261 [07:27<00:05, 2.75it/s]
99%|| 1247/1261 [07:28<00:04, 2.81it/s]
99%|| 1248/1261 [07:28<00:04, 2.76it/s]
99%|| 1249/1261 [07:28<00:04, 2.83it/s]
99%|| 1250/1261 [07:29<00:03, 2.78it/s]
99%|| 1251/1261 [07:29<00:03, 2.83it/s]
99%|| 1252/1261 [07:29<00:03, 2.78it/s]
99%|| 1253/1261 [07:30<00:02, 2.84it/s]
99%|| 1254/1261 [07:30<00:02, 2.77it/s]
100%|| 1255/1261 [07:30<00:02, 2.83it/s]
100%|| 1256/1261 [07:31<00:01, 2.76it/s]
100%|| 1257/1261 [07:31<00:01, 2.82it/s]
100%|| 1258/1261 [07:32<00:01, 2.76it/s]
100%|| 1259/1261 [07:32<00:00, 2.83it/s]
100%|| 1260/1261 [07:32<00:00, 2.79it/s]
```

```
[MoviePy] Done.
[MoviePy] >>> Video ready: project_video_output.mp4
```

```
CPU times: user 4min 12s, sys: 1min 14s, total: 5min 27s
Wall time: 7min 35s
```

```
In [30]: from IPython.display import HTML
```

```
HTML("""<video width="480" height="270" controls><source src="{0}">
</video>""").format(video_output1))
```

```
Out[30]: <IPython.core.display.HTML object>
```

Briefly discuss any problems / issues you faced in your implementation of this project. Where will your pipeline likely fail? What could you do to make it more robust?

The main issue is to locate the right source and goal focuses. It is a hit and preliminary methodology and even 5 pixels all over can have a major effect. The subsequent issue is the point at which I was attempting to utilize different blends of shading channels the last mix didn't work in practically all conditions. It was again by hit and preliminary I made sense of awful casings and checked my pipeline and made changes to as well as administrators and limits. The following test and the most serious issue is to quit flashing of path lines on solid surface or when the vehicle comes out from the shadow.

I gave my pipeline a shot the test video and I saw it fizzled. So I will explore different avenues regarding the test video without a doubt. It is very conceivable that left path line to focus is of various shading and from focus to right path is of various shading as in the test video and it is probably going to come up short there. Likewise if there should arise an occurrence of a mountain landscape, it is very prone to fall flat.

To make it progressively powerful and stop the flashing of path lines, we can average out the focuses from the past casings to have a smooth change for each edge.