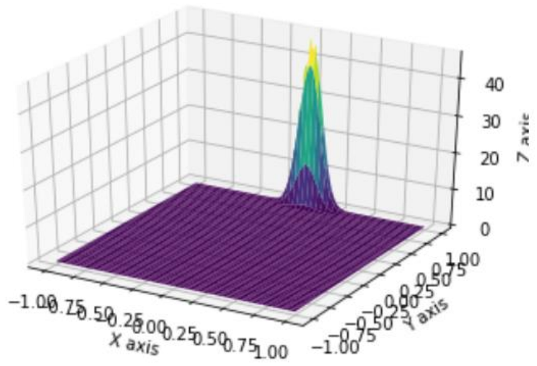
CSE514A, 2019Fall **CourseProject2 Report** **Xiaoyao Wang 465943**

Intro

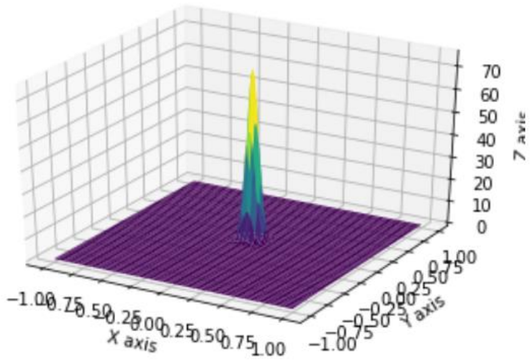
This report follows step by step, aiming to answer specifically the required questions in the instruction.

Step 2.1.2

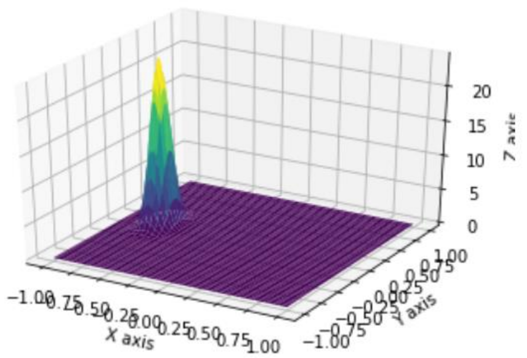
Normal distribution for ***V0***



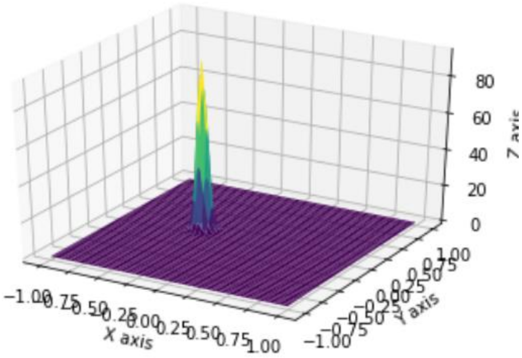
Normal distribution for ***V0'***



Normal distribution for ***V1***



Normal distribution for ***V1'***



Comparing V0 with V0' and V1 with V1', we can observe that the width of 3-D normal distribution of Vk' is smaller than that of Vk, indicating that Vk' has a lower variance in its latent variables than Vk. Another observation is that the distribution of Vk' is closer to the original point of the x-y plane, meaning that the means of the latent variables of Vk' is closer to zero than that of Vk.

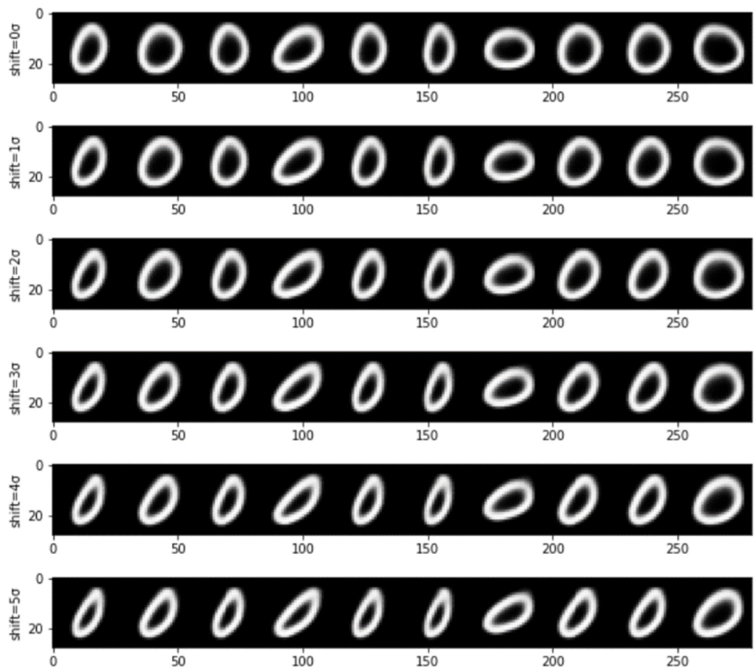
After running the Vk and Vk' for several times, I also observed that the shape of Vk' (distribution) is more stable than that of Vk. To be more specific, the two latent variables of Vk could sometimes be very different. For instance, the plot of V0 above shows that one variable has a high variance and its mean is close to zero while the other variable has low variance with its mean away from zero. When I run another time the distribution of V0 could then become similar to the distribution of V1 above. On contrast, the shape of Vk' does not vary much and it looks like a narrow stick most of times.

The last interesting thing is that the shape of Vk, as discussed in last paragraph, could be compressed in in one dimension but wide in the second dimension sometimes. I think such phenomenon may tell that the features of a certain hand-written digit can be decently encoded using only one latent dimension.

Generally, the distribution of Vk' is more concise and stable than that of Vk as it is trained using a much larger data sample. Also for one digit, its features could be decently represented using one dimension rather than two.

Step 2.2.2

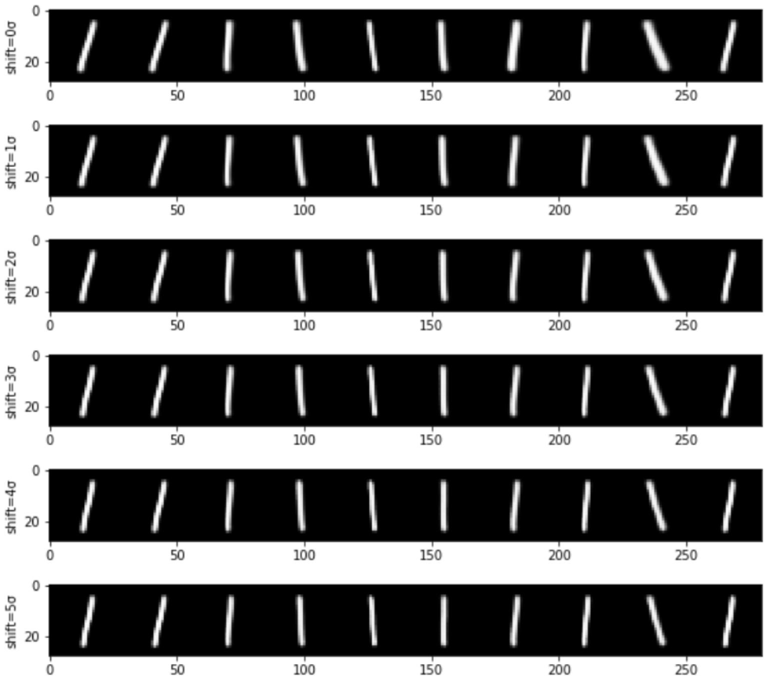
10 test images of V0	
Mean Vector	**Log_sigma Vector**
[[0.13528046 0.5303803]	[[-3.7554932 -3.6405196]
[-0.19223487 -0.46684432]	[-3.8750234 -3.5041971]
[-0.32193768 0.08100079]	[-3.8997288 -3.5437508]
[0.6057198 -0.89939964]	[-3.4846995 -3.3991766]
[-0.13630234 0.56923276]	[-3.7522633 -3.5393715]
[-0.07112733 1.0933996]	[-3.7958224 -3.667018]
[0.8407048 -2.341048]	[-3.511672 -3.3709083]
[-0.2631055 -0.6223335]	[-3.7141945 -3.3125958]
[-0.2593185 -0.32042858]	[-3.7791307 -3.409482]
[-0.6711788 -1.8592303]]	[-4.385065 -3.6397185]]
Images with shift= δ *i for δ =0.25 and i = 0,1,2,3,4, and 5	



10 test images of V1

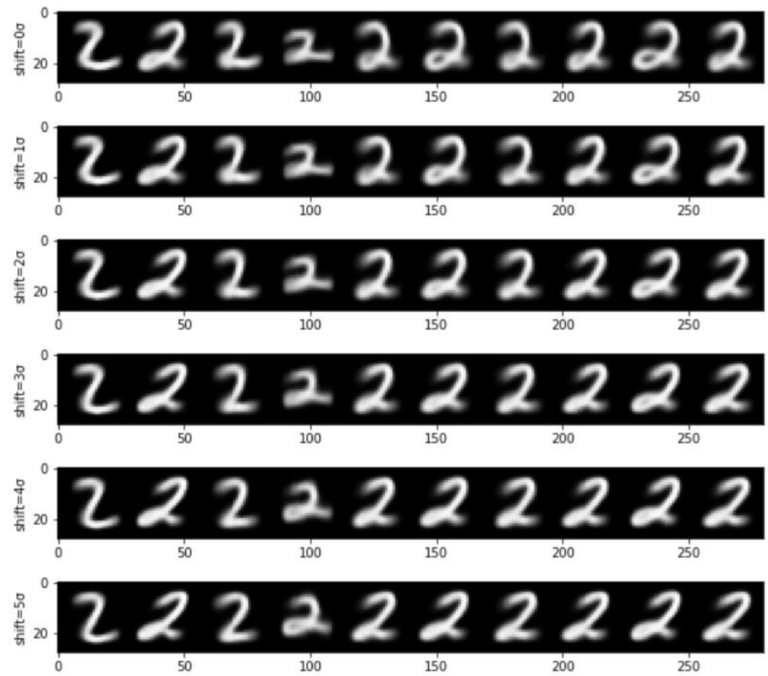
Mean Vector	Log_sigma Vector
[[0.3439393 1.3975941]	[[-3.8409336 -3.214809]
[0.4204387 1.5942433]	[-3.6703267 -3.0158458]
[0.43981475 -0.10019933]	[-3.7904575 -3.5919523]
[1.3295364 -0.5755755]	[-2.821185 -3.0683675]
[3.6259553 0.20011586]	[-1.5125632 -3.0084105]
[1.1432893 -0.22811478]	[-2.9239419 -3.1822777]
[0.25487357 -0.18712473]	[-3.7770364 -3.397756]
[1.0057098 0.52231103]	[-3.2352543 -3.3578258]
[1.6092784 -2.4048736]	[-2.658478 -2.6691275]
[0.6136445 1.3023593]]	[-3.9105027 -3.3950233]]

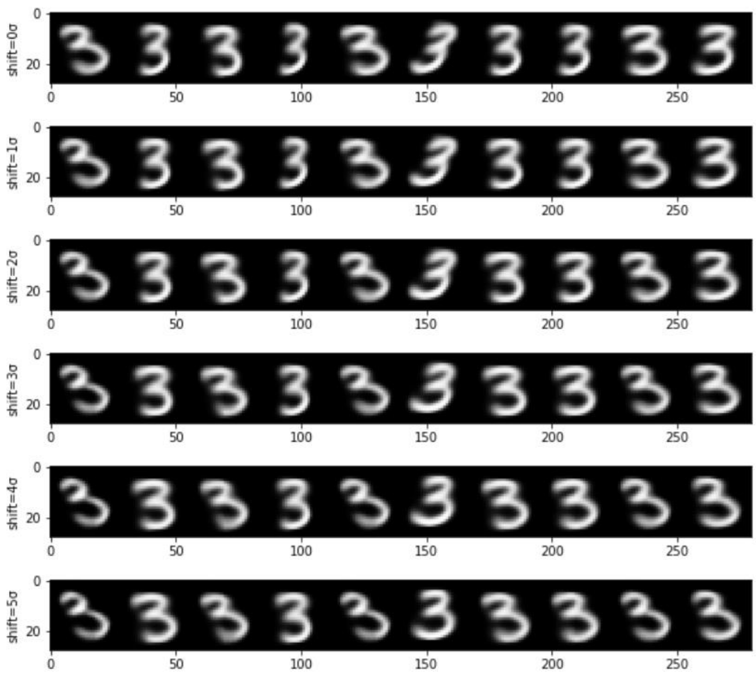
Images with shift= δ *i for δ =0.25 and i = 0,1,2,3,4, and 5



10 test images of V2	
Mean Vector	**Log_sigma Vector**
[[-3.7334177 -0.03879758]	[[-2.1457407 -3.2930346]
[1.5108697 0.5021235]	[-2.487207 -3.47846]
[-1.7234696 -0.1582902]	[-2.8344128 -3.5445564]
[-1.1105211 -2.2472725]	[-2.742846 -3.5142727]
[-0.2007287 0.02333728]	[-3.0309389 -2.9234965]
[0.6547533 -0.09137829]	[-3.0068185 -3.443397]
[-0.42415312 0.01817618]	[-3.158346 -3.673337]
[0.49150214 0.28445056]	[-3.2400157 -3.3392394]
[1.2977505 -0.22429413]	[-3.2203119 -3.7228684]
[0.37537906 0.35679123]]	[-3.0779328 -3.404147]]

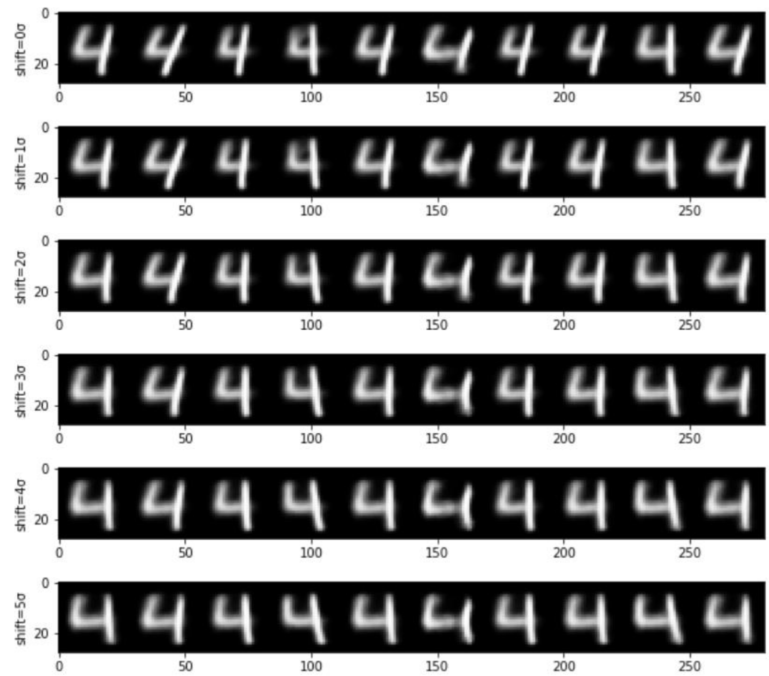
Images with shift= δ *i for δ =0.25 and i = 0,1,2,3,4, and 5



<table>
<tr><th colspan="2">10 test images of V3</th></tr>
<tr><th>Mean Vector</th><th>Log_sigma Vector</th></tr>
<tr><td>[[0.95571595 1.8160288]
 [-1.1609265   0.23937622]
 [-0.5363634   1.282897]
 [-1.8854247 -0.2544549]
 [0.37506875 1.4333837]
 [-0.72669625 -1.6032467]
 [-0.8218734   0.45914394]
 [-0.7588873   0.1494448]
 [0.1452621   0.920373]
 [-0.1099157 -0.00674739]]</td><td>[[-3.3521113 -2.6496904]
 [-2.6169853 -2.6589792]
 [-2.9726684 -2.709097]
 [-1.7166994 -2.31803]
 [-3.4580686 -2.6407566]
 [-3.2052543 -4.1438804]
 [-3.06204   -2.9025292]
 [-2.7818468 -2.727174]
 [-2.7754748 -2.4708698]
 [-3.5575855 -3.1546698]]</td></tr>
<tr><th colspan="2">Images with shift= δ *i for δ =0.25 and i = 0,1,2,3,4, and 5</th></tr>
<tr><td colspan="2">
</td></tr>
</table>

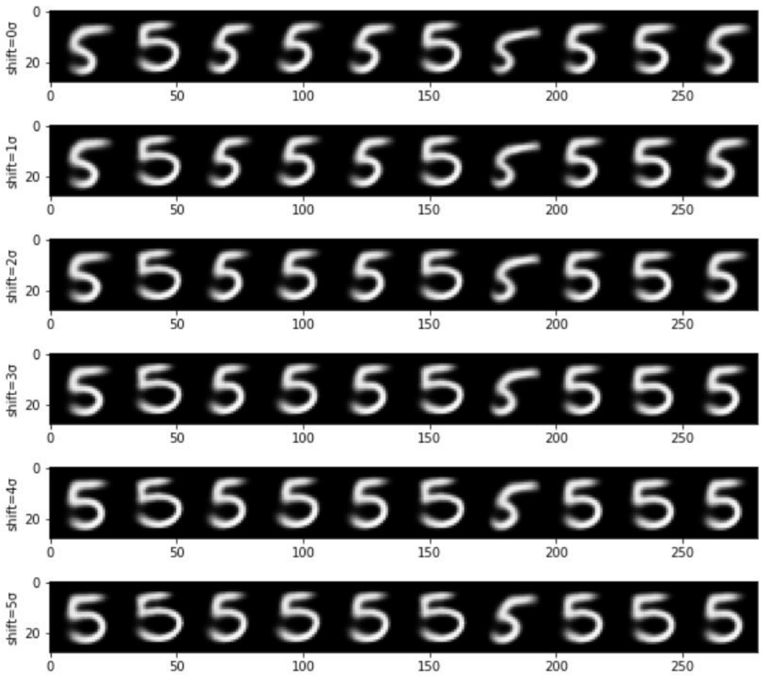
10 test images of V4	
Mean Vector	**Log_sigma Vector**
[[0.25380832 0.74739397] [-0.39031416 0.37584966] [0.13927534 -0.22732209] [0.7393315 -0.38427895] [0.166549 0.43889484] [0.4417672 3.2753115] [-0.02728448 -0.25930458] [0.09786046 0.38273796] [0.88729036 0.84835947] [0.1377877 1.177026]]	[[-3.2834277 -3.4750507] [-3.4486365 -3.2029605] [-3.3760667 -3.3942232] [-3.1156726 -3.195629] [-3.3626847 -3.4714518] [-3.9182277 -2.6405246] [-3.451291 -3.3665195] [-3.363793 -3.4225605] [-3.3134332 -3.2843251] [-3.292264 -3.4586782]]

Images with shift= δ *i for δ =0.25 and i = 0,1,2,3,4, and 5

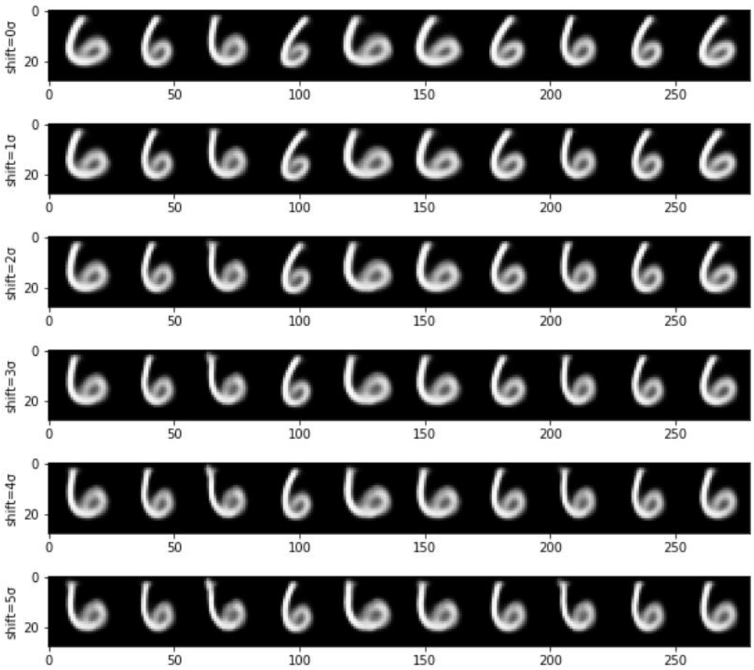


10 test images of V5	
Mean Vector	**Log_sigma Vector**
[[-1.2282621 0.16224156]	[[-2.8265085 -3.115852]
[1.1878616 1.7281119]	[-2.5898874 -3.2245111]
[-0.17123748 -0.11728835]	[-3.1632044 -3.2853453]
[0.26777998 0.18450417]	[-3.120359 -3.3562388]
[-0.24288994 -0.11409269]	[-3.0285003 -3.2505777]
[0.7159554 1.3074932]	[-2.8186295 -3.4243188]
[-1.5051506 -1.5516951]	[-2.138152 -2.4724064]
[-0.3975202 0.48419303]	[-3.014023 -3.3227844]
[-0.5038389 0.85764647]	[-2.8491676 -3.3333642]
[-0.7856851 -0.09312553]]	[-3.016708 -3.1977813]]

Images with shift= δ *i for δ =0.25 and i = 0,1,2,3,4, and 5

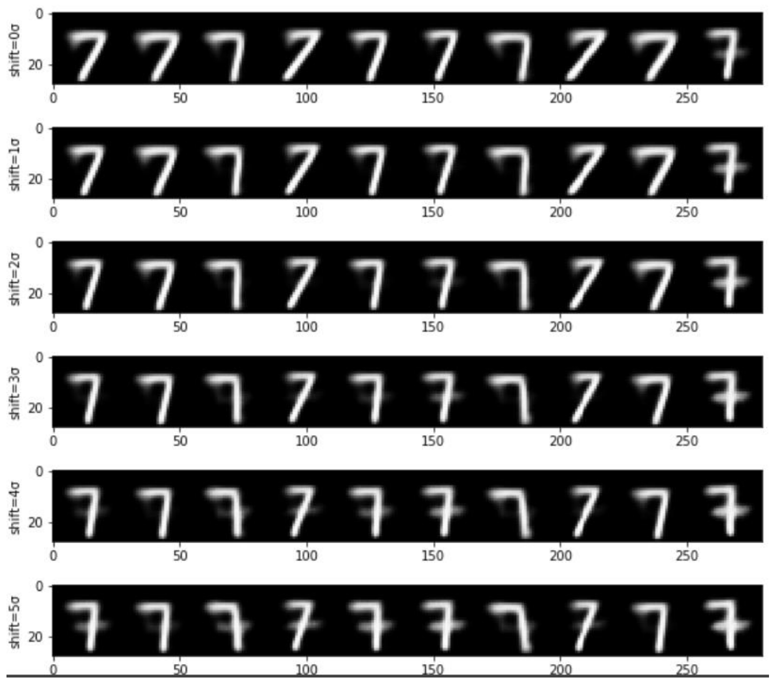


10 test images of V6	
Mean Vector	**Log_sigma Vector**
[[0.05044498 1.112513]	[[-2.9902241 -3.34741]
[0.72867787 0.02304665]	[-3.12255 -3.3359258]
[0.9287569 1.8498125]	[-2.8213675 -3.2014635]
[-0.00865443 -1.0253971]	[-3.1180203 -3.2264366]
[-0.3696343 2.1584268]	[-2.7682874 -3.0092611]
[-0.12605585 1.1999061]	[-3.0180857 -3.46254]
[0.14007655 -0.10074139]	[-3.3444953 -3.6717947]
[0.8525248 0.9008028]	[-3.1273196 -3.3025851]
[0.4265754 -0.30771396]	[-3.0553622 -3.3633237]
[-0.09570861 -0.03090787]]	[-3.144312 -3.4875994]]
Images with shift= δ *i for δ =0.25 and i = 0,1,2,3,4, and 5	

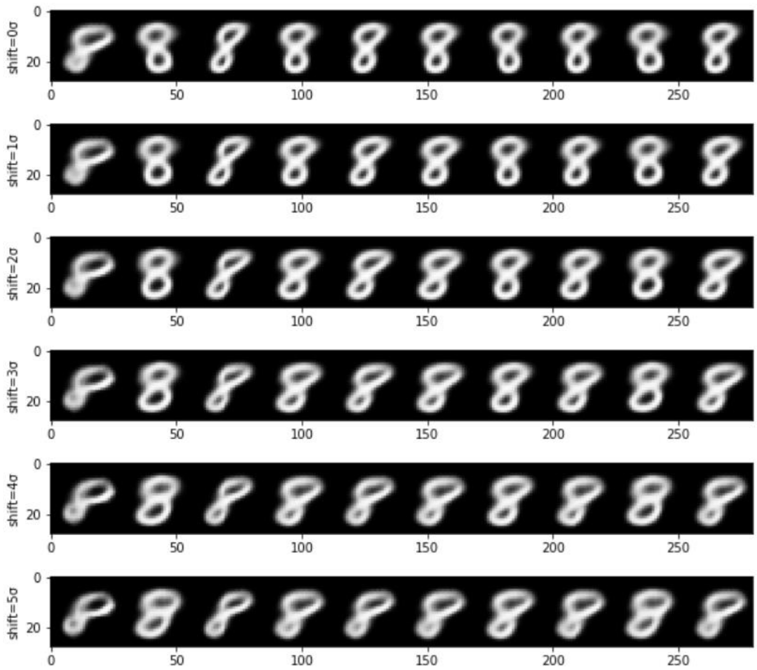


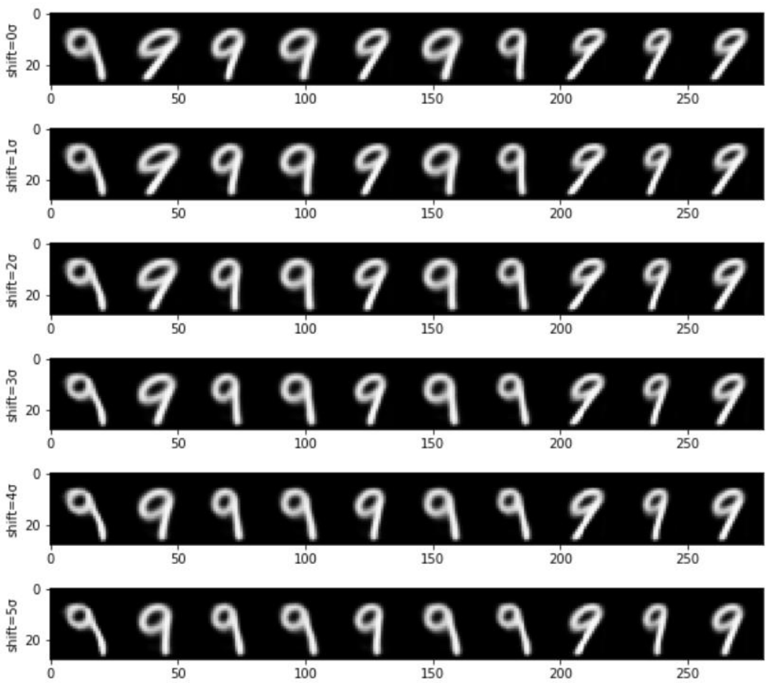
10 test images of V7	
Mean Vector	**Log_sigma Vector**
[[-1.6123453e-01 -1.2246382e-01]	[[-2.9737358 -2.9929314]
[3.9185952e-02 -6.9748998e-01]	[-3.162634 -2.9800868]
[1.2394649e+00 -5.5492353e-01]	[-3.38245 -2.7502456]
[-1.2699488e+00 5.8586276e-01]	[-3.2399178 -2.8884459]
[3.7435392e-01 -7.2126389e-02]	[-3.5480843 -3.4942756]
[1.6122658e-03 4.7433689e-01]	[-3.420278 -3.11097]
[1.7578198e+00 -1.1713926e+00]	[-3.6876087 -2.8047237]
[-1.5653396e+00 3.1987303e-01]	[-3.6558208 -2.8267117]
[-1.5369704e-01 -9.8006338e-01]	[-3.208972 -2.876031]
[7.6134956e-01 1.2124826e+00]]	[-3.0929918 -2.8419113]]

Images with shift= δ *i for δ =0.25 and i = 0,1,2,3,4, and 5



10 test images of V8	
Mean Vector	**Log_sigma Vector**
[[1.0176888 1.2843943] [-1.0060322 0.04005384] [0.508958 -0.35908] [-0.4642345 -0.12468684] [0.03996382 -0.25546777] [-0.32764244 -0.09221125] [-0.7741725 -0.41226393] [-0.4998339 -0.06333756] [-0.9457642 0.01505537] [-0.2129812 -0.1483433]]	[[-3.0889316 -3.2448475] [-3.6630435 -3.5498598] [-3.6919668 -3.2797701] [-3.6286466 -3.3315754] [-3.6376548 -3.2767978] [-3.674837 -3.3638697] [-3.2425706 -3.092375] [-3.3876226 -3.1684685] [-3.2685182 -3.205121] [-3.3237348 -3.0527344]]
Images with shift= δ *i for δ =0.25 and i = 0,1,2,3,4, and 5	



10 test images of V9	
Mean Vector	**Log_sigma Vector**
[[1.8915594 0.88668716]	[[-4.105055 -3.2558222]
[-0.87699586 -0.833545]	[-3.314133 -3.2709746]
[-0.32823053 0.17539091]	[-3.3444376 -3.2290888]
[0.14253922 0.02600684]	[-3.6593328 -3.4252338]
[-1.319042 -0.3188333]	[-3.216273 -3.1753292]
[0.08158777 -0.1147772]	[-3.5865457 -3.3921623]
[-0.134753 0.99927056]	[-2.991219 -2.817748]
[-2.3757236 -1.544324]	[-3.0023699 -2.8892941]
[-2.3845236 0.2416302]	[-3.0016744 -2.895115]
[-2.109008 -0.9109855]]	[-2.9276035 -2.830214]]
Images with shift= δ *i for δ =0.25 and i = 0,1,2,3,4, and 5	
	

When observing the above images for k=0,1,2,3,4,5,6,7,8, and 9, δ =0.25, and i = 1,2,3,4, and 5, we can see that for all ks, the **clarity** of output images does not change much when i changes from 1 to 5. However, there is larger **distortion** of output images when i changes from 1 to 5, i.e. with more shift added to the mean vector. For any Vk, as more shifts are added to the mean vector, they all become **more different from the original** VAE output images with no shift in mean vector. Larger shift will *weaken the features (mean vector) that are extracted from the input images and will be sent to the decoder*, and hence deteriorate the output image quality.

Another finding is that the original 10 output images without shift could be very different from each other, but as more shifts are added to the mean vector, the 10 images becomes **more similar to each other**, indicating that larger shift will *dominate the information sent to the decoder*, and hence deteriorate the output image quality.

However, overall, the modified output images are **still very similar to** the original output ones. A small shift in the embedding space does not deteriorate the output images much.

Step 2.2.3

2 similarity functions are used:

1) Mean Squared Error

$$MSE = \frac{1}{m\,n}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2$$

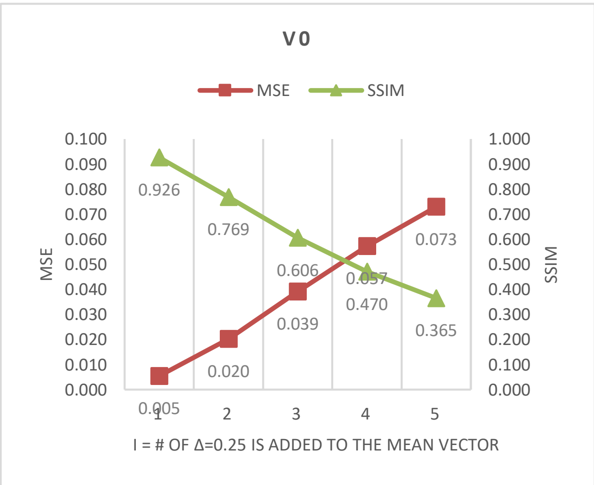
where m,n respectively represent the number of pixel rows and the number of pixel columns for both image I and K.
A value of 0 for MSE indicates perfect similarity. A value greater than one implies less similarity.

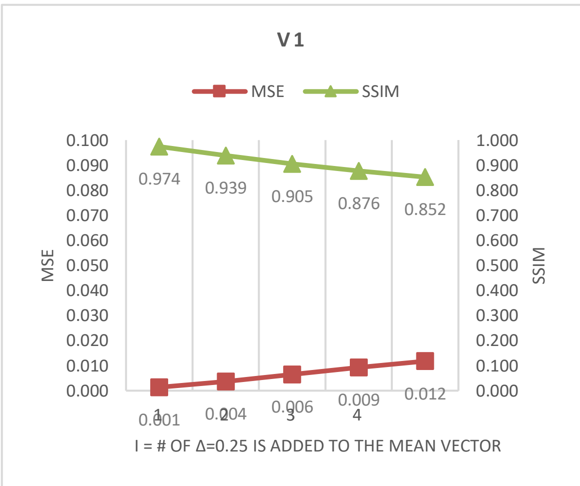
2) Structural Similarity Index

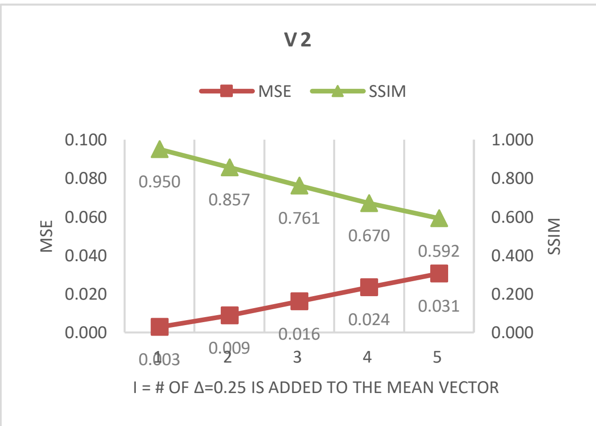
$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

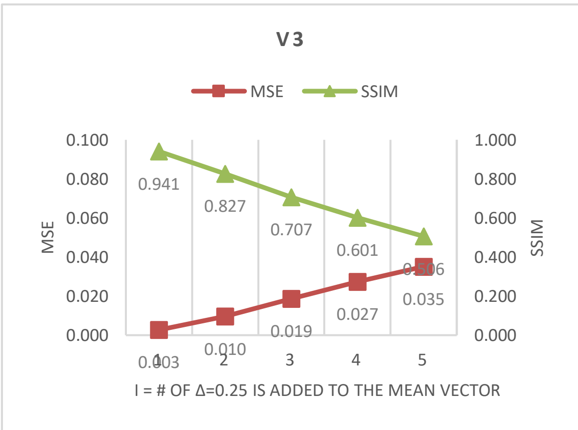
The parameters include the (x, y) location of the N x N window in each image, the mean of the pixel intensities in the x and y direction, the variance of intensities in the x and y direction, along with the covariance.
The value can vary between -1 and 1, where 1 indicates perfect similarity.

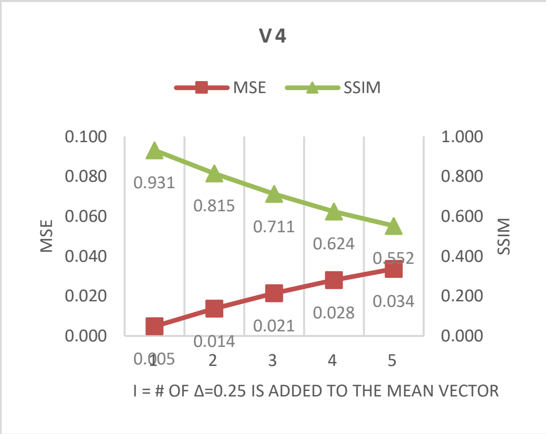
With δ =0.25:

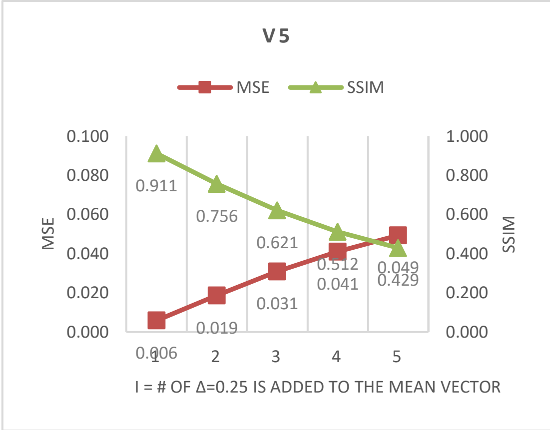


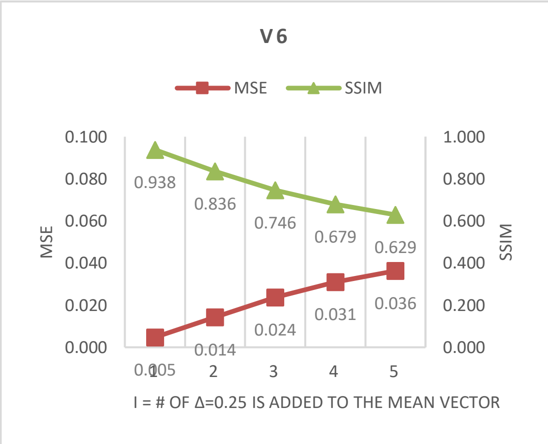


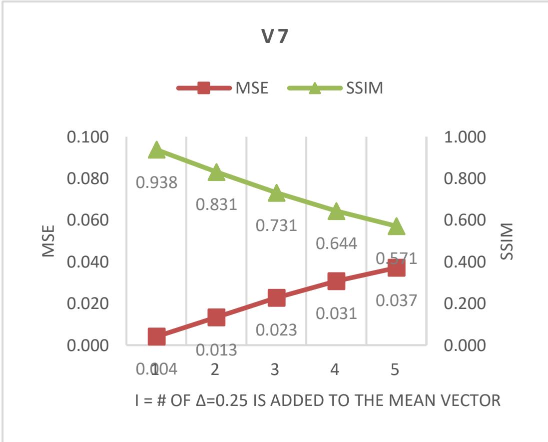


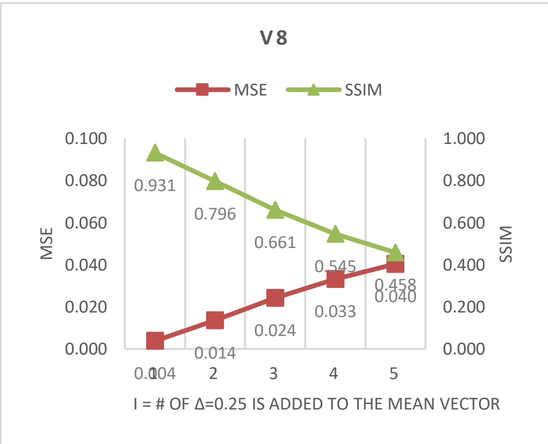


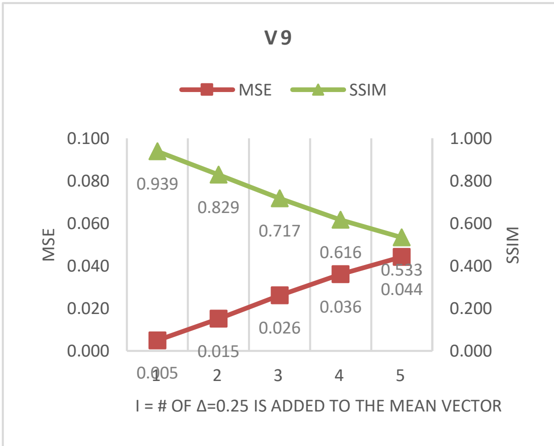












As i changes from 1 to 5, MSE increases and SSIM decreases linearly, indicating that the similarity decreases linearly with more shifts added to the mean vector. In other words, there is a linear relationship between the shift added to the latent variable and the degree of distortion of output images. If we think about the 3-D normal distribution graph, i* δ should only change the position of the distribution by (i* δ , i* δ) on the x-y plane rather than the shape. Maybe this can explain why the degree of distortion is linear to the degree of shift in mean vector.

One thing notable is that the absolute value of the slope of V0's similarity plot is much larger than that of V1's similarity plot. It seems that i* δ has more effect on V0 than V1. Such phenomenon corresponds to what we observed visually from the 100 images in Step 2.2.2.

Maybe we can explain this situation by looking into the nature of the shapes of digits 0 and 1, i.e. digit 1 has the simplest shape among all the 0~9 digits. Therefore the i* δ added to the latent variables has more effect on V0 which contains more information in latent space than V1.

Step 2.2.4

I think that the decoder would **not** be exactly consistent with the revised model. If the change in mean or standard deviation is a shift along the vector, then the decoder learned from learning the model Vk could still generate some reasonable images, but if the change is similar to some random noise then the original decoder can not perform well.

In terms of the 'shift', step 2.2.2 shows that a small shift in mean vector would lead to some degree of distortion in output images from the original decoder, though shift will not change the shape of the normal distributions of latent variables but their positions. As the normal distributions move, any point on distributions will have a different coordinate value, and thus containing different information. If we put the output of the revised encoder to the original decoder, every pixel would move a bit away from its original output position when we use the output of the original encoder. Maybe that's why we observe that the output images of step 2.2.2 tend to lean to left or right as the shift becomes larger. In this case we can still get something that looks like the original outputs.

I think the revised decoder would adjust its parameters to learn the changes in mean or standard deviation automatically just like DAE does so it would not be consistent with the original decoder.