Islam Gamal	11
Karim Mohamed	45

Search by Images

Using Auto Encoders

The selected one to expand on.

Feature Detection

Using algorithms for matching image deformation such as blur, rotation, scale, and illumination change.
Such as **SIFT**, **PCA-SIFT** and **SURF**

Search by Color

"Piximilar Multicolor Search process"

Given archive of images→ create digital signature based colors (color histograms) → store the extracted signatures in database → whenever given a query image, extract its colors → find the most suitable matching image

Feature Detection

A set of algorithms is used to analyze image attributes ex (colour, shape, texture, luminosity, complexity, objects and regions). These attributes are stored for indexing and this can be used with keywords to refine searches on extremely large collections.

Auto Encoder

Encoding Images into 1D vector array and then finding the Nearest neighbours to a given query image.

Related work.

People Approaches to the Auto encoder.

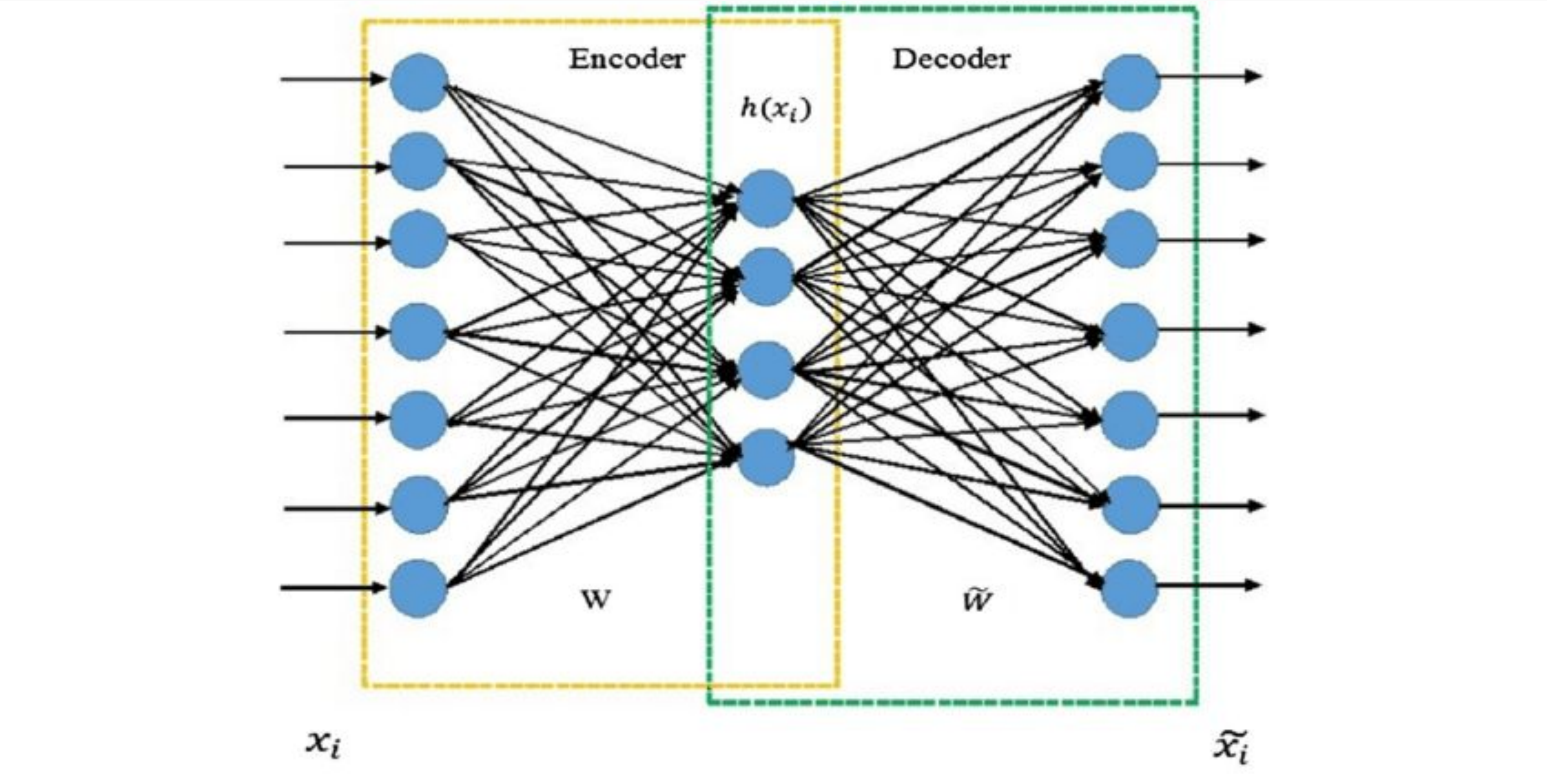
1. Fully connected layers.

2. Fully convolutional way
   a. Encoder : Conv + pooling
   b. Decoder : Conv + Up sampling

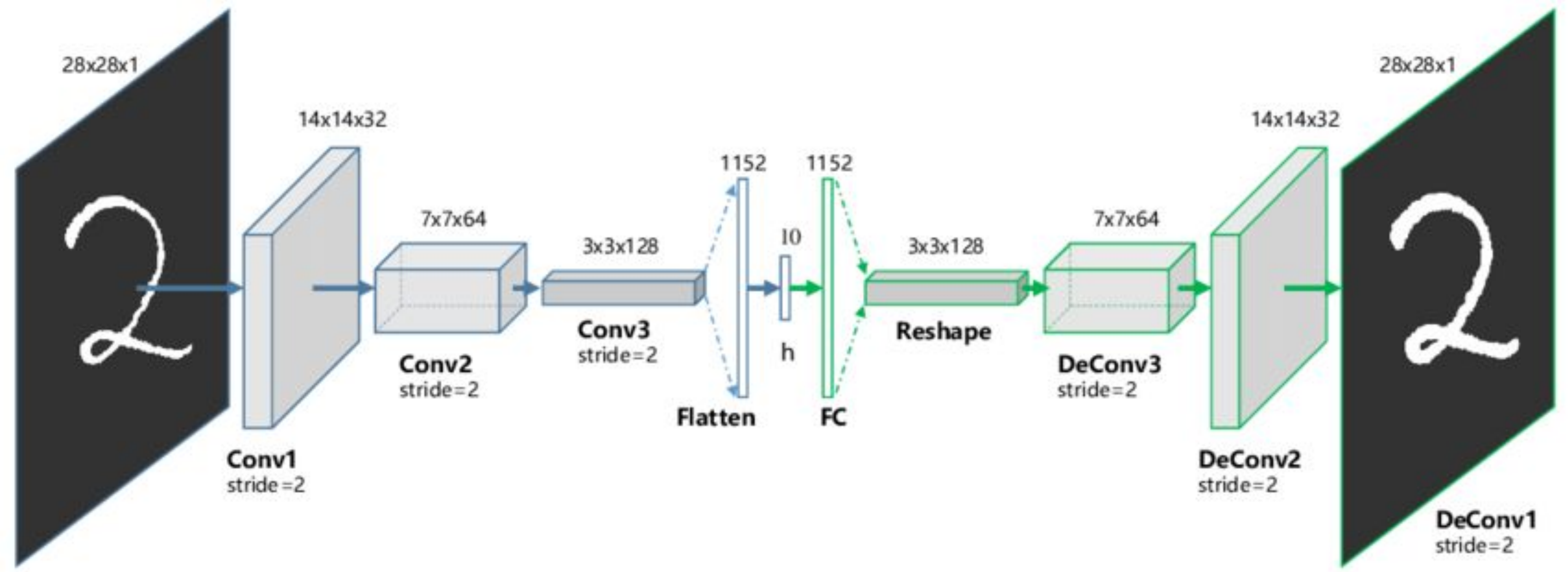
3. Convolution , DeConvolution
   a. Encoder : Accumulative convolutions layers + Pooling.
   b. Decoder : Accumulative transpose convolutional layers.

4. Mix of 2 and 1

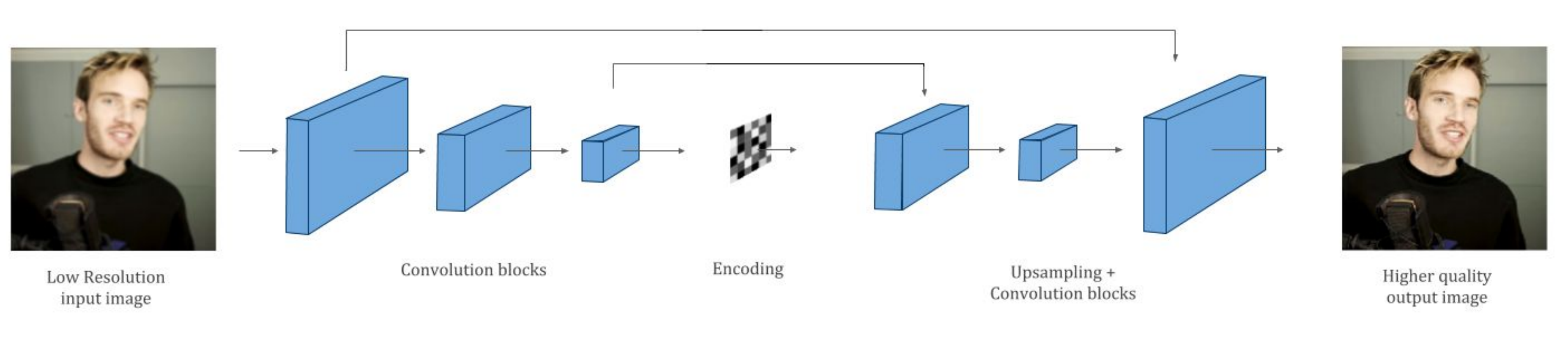
Fully Connected Layers Methodology - Architecture



Fully Convolutional way - Architecture



Convolutional + Up Sampling - Architecture

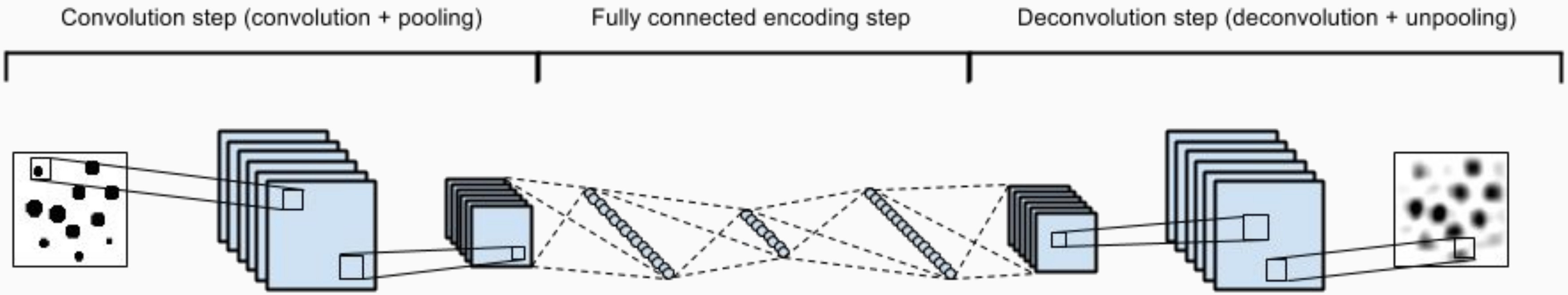


Our proposed solution/model and Difference from related work.

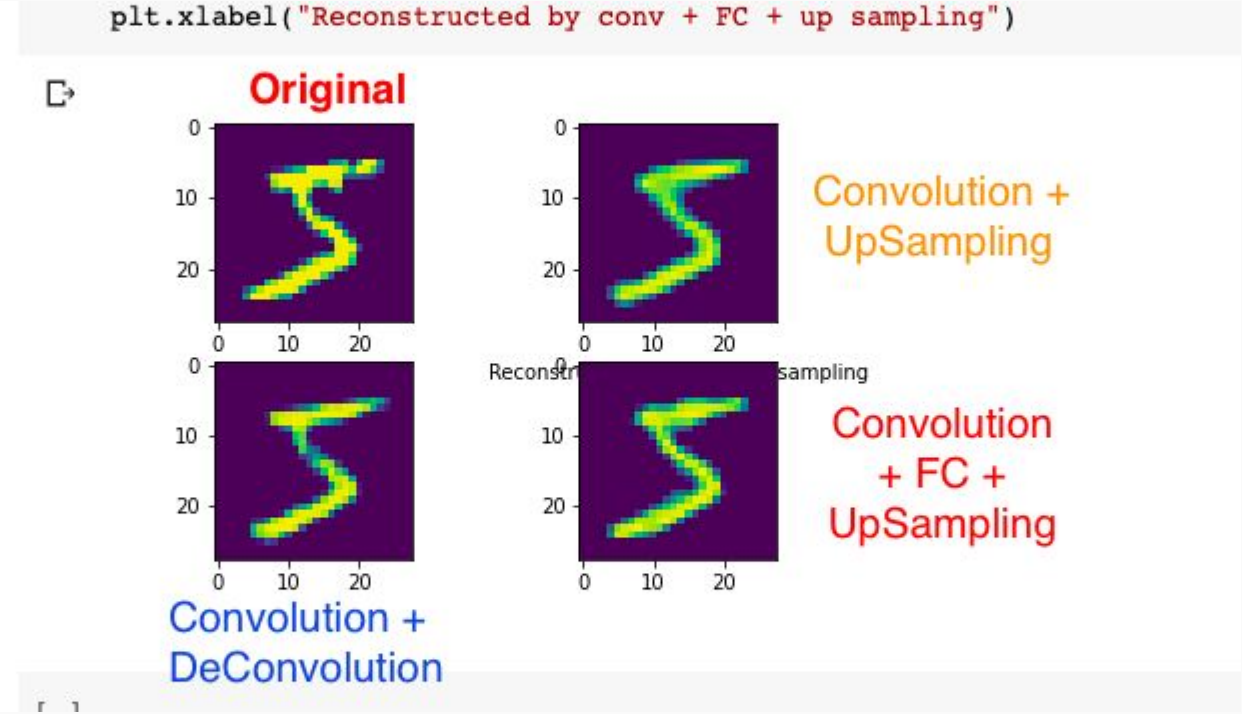
mixed approach with Convolutional Neural Networks (CNNs) and FC layers

- ➢ The **Encoder** consists of the following layers
  - ✓ Convolutional Layers.
  - ✓ Non-Linearity layers.
  - ✓ Rectification Layers (can use just ReLU).
  - ✓ Pooling layers (Average and Max pooling)
  - ✓ Fully connected layers.
  - ✓ Dropout layers.
- ➢ The **Decoder** consists of the following layers
  - ✓ FC layers (undos effect of last layer in encoder).
  - ✓ Transpose Convolution Layers (Reversing the encoder)

Model Architecture (CNN + FC + Trans Conv)



Applying the models on MNIST (Handwriting) dataset.



Approach \ MSE train loss by	Them	Winner	Us
Fully Connected Only	0.0214	→	0.0026
Convolution + Up Sampling	0.0407	→	0.0065
Convolution + DeConvolution	0.0297	→	0.0091
Conv + FC + Up Sampling	0.0173	→	0.0050

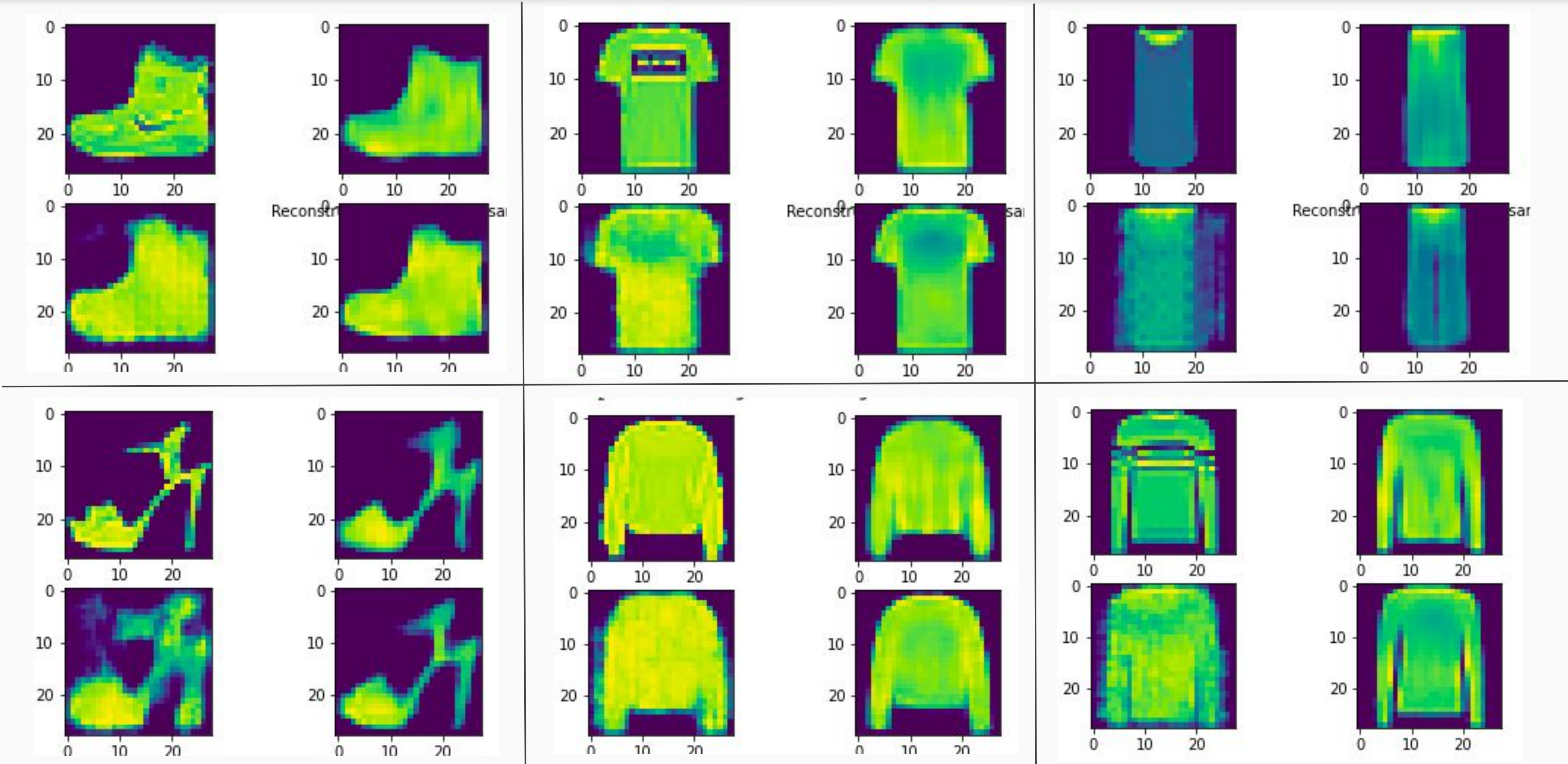
Them :
https://towardsdatascience.com/aligning-hand-written-digits-with-convolutional-autoencoders-99128b83af8b
https://towardsdatascience.com/build-a-simple-image-retrieval-system-with-an-autoencoder-673a262b7921
https://medium.com/analytics-vidhya/building-a-convolutional-autoencoder-using-keras-using-conv2dtranspose-ca403c8d144e

Confusing

Using a new dataset

MNIST for fashion

Applying the models on MNIST (Handwriting) dataset.



Approach \ MSE train loss by	Them	Us (Fashion MNIST)	Us (HandWriting MNIST)
Fully Connected Only	0.0214	0.00543	0.0026
Convolution + Up Sampling	0.0407	0.0131	0.0065
Convolution + DeConvolution	0.0297	0.049	0.0091
Conv + FC + Up Sampling	0.0173	0.0091	0.0050

Them :
https://towardsdatascience.com/aligning-hand-written-digits-with-convolutional-autoencoders-99128b83af8b
https://towardsdatascience.com/build-a-simple-image-retrieval-system-with-an-autoencoder-673a262b7921
https://medium.com/analytics-vidhya/building-a-convolutional-autoencoder-using-keras-using-conv2dtranspose-ca403c8d144e

VAE

Variational Autoencoders

2. Using Variational autoencoders instead on normal autoencoders.

WHY?

Given a dataset that we want to encode it all, the resulted latent layers will be in a clustering form, not a continuous form, so if a target image contained some noise or error , there is a high probability that the produced vector will not be close to the original one , it may be considered as another cluster.

(Unsupervised learning)

VAE

Variational Autoencoders

2. Using Variational autoencoders instead on normal autoencoders.

BUT

With using Variational autoencoders, the encoded dataset will be in a continuous way since the latent layers has learnt a distribution in the input images.

So it will be more reasonable and confidential to use the nearest neighbour algorithms to find the target image from it's encoded version.

(Supervised learning)

VAE Model Architecture. (encoder)

```
Model: "encoder"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
encoder_input (InputLayer)      (None, 784)          0
__________________________________________________________________________________________________
dense_4 (Dense)                 (None, 512)          401920      encoder_input[0][0]
__________________________________________________________________________________________________
z_mean (Dense)                  (None, 2)            1026        dense_4[0][0]
__________________________________________________________________________________________________
z_log_var (Dense)               (None, 2)            1026        dense_4[0][0]
__________________________________________________________________________________________________
z (Lambda)                      (None, 2)            0           z_mean[0][0]
                                                                 z_log_var[0][0]
==================================================================================================
Total params: 403,972
Trainable params: 403,972
Non-trainable params: 0
```

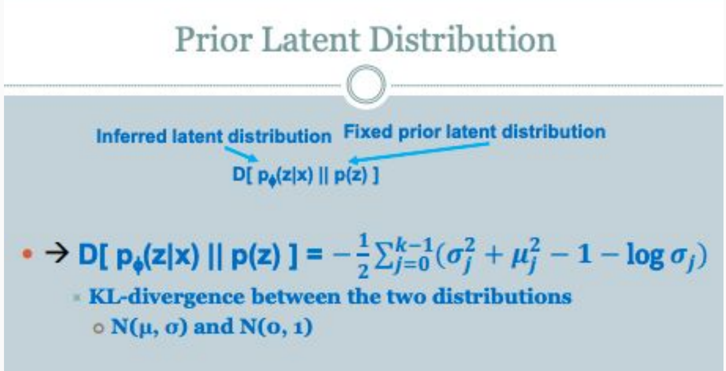
VAE Model Architecture. (decoder)

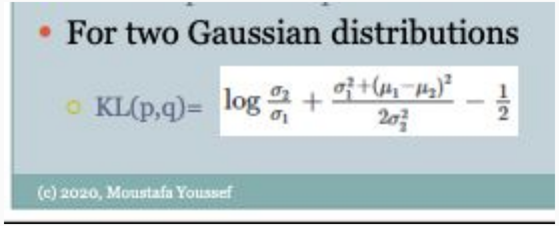
```
Model: "decoder"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
z_sampling (InputLayer)      (None, 2)                 0
_________________________________________________________________
dense_5 (Dense)              (None, 512)               1536
_________________________________________________________________
dense_6 (Dense)              (None, 784)               402192
=================================================================
Total params: 403,728
Trainable params: 403,728
Non-trainable params: 0
```

Evaluation Metrics

Reconstruction loss + K-L loss.

- Reconstruction loss → using (Cross Entropy) OR (mean square error)
- K-L loss → K-L Divergence Metric → rule.



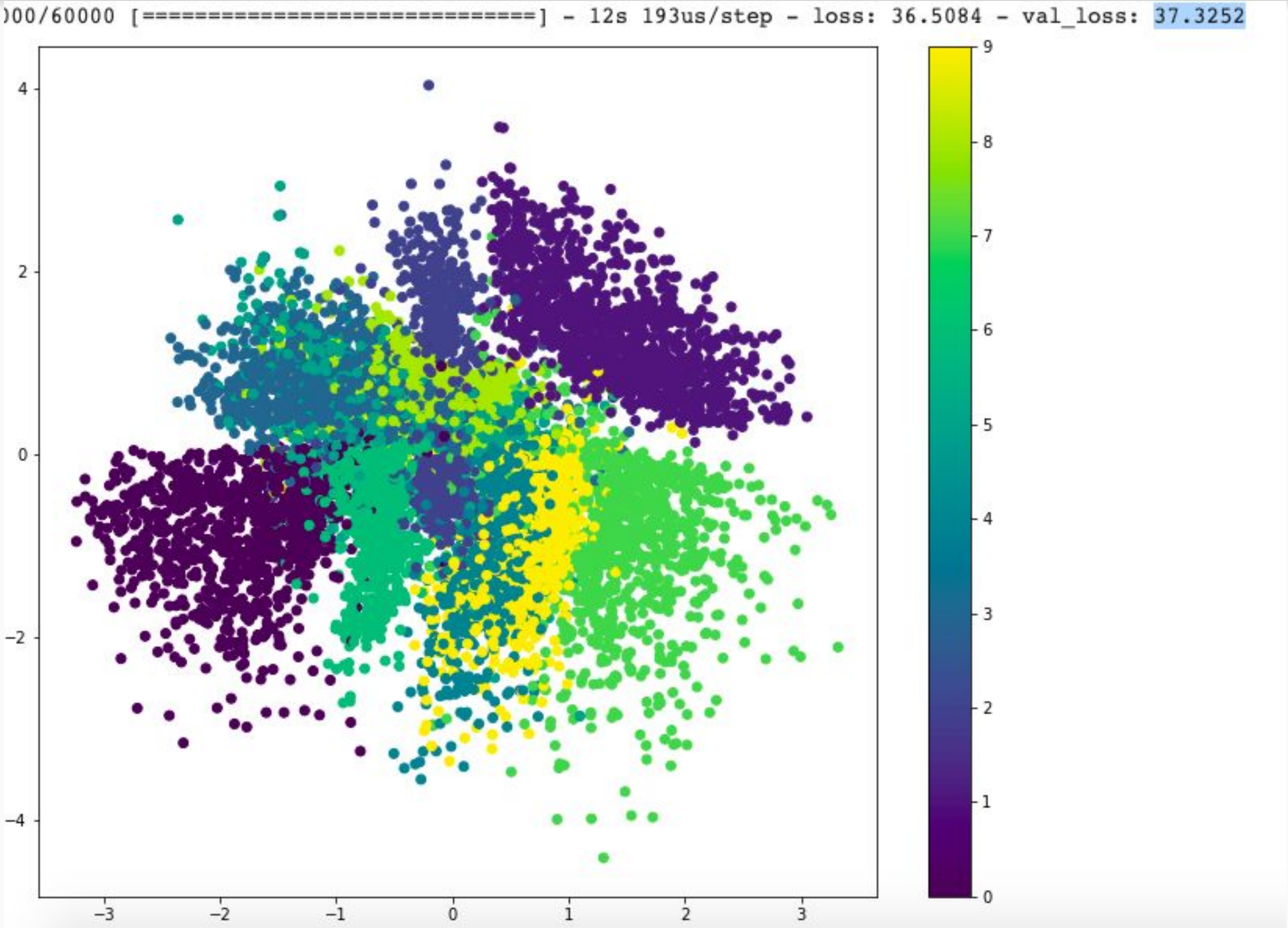


From Dr. Moustafa Youssef Slides
https://piazza.com/class_profile/get_resource/k6mquuyyw903ch/k9cp4lzsi2a1n8

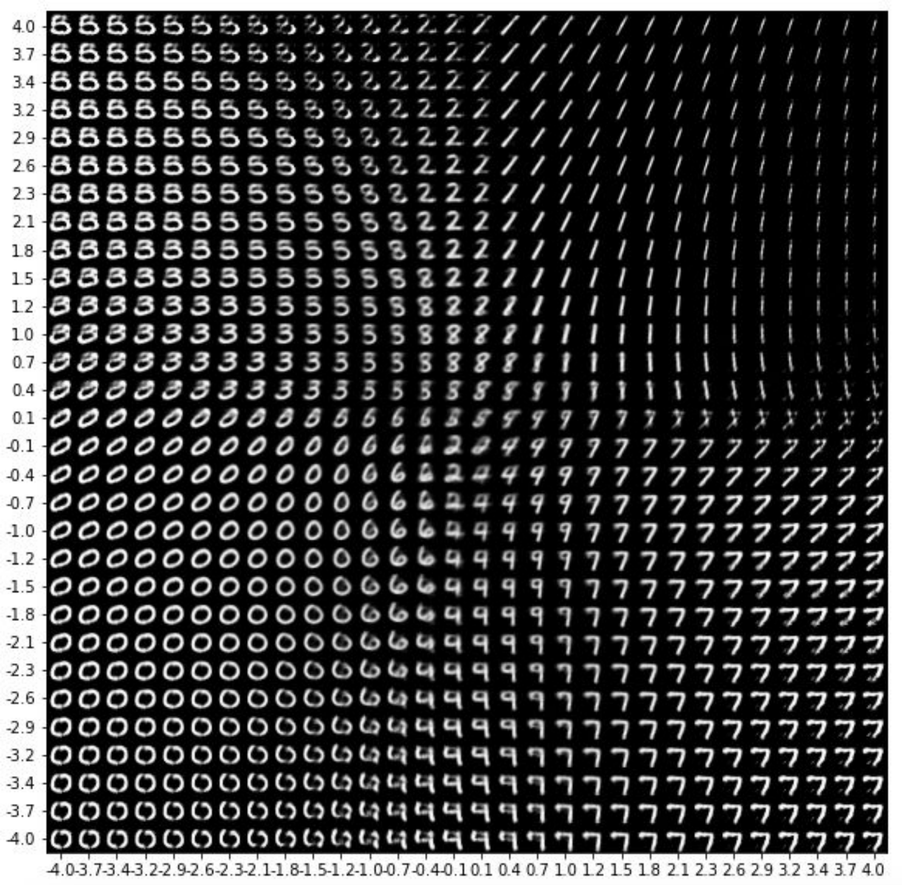
37.3252

Evaluated Loss

Visualizations to the continuous distribution of the latent vectors.



Visualizations to the continuous distribution of the latent vectors. (Interpolation)



Another proposed solution/model and Difference from related work.

3. Third we will try to improve that by using Recurrent Neural Networks (**RNN**) instead of CNN:

   ➢ LSTM network:

   ✓ Each layer of LSTM has as many cells as the timesteps and using ReLU as activation function.

   ✓ It takes 2d as input (each layer).

   ✓ If the subsequent layer is also LSTM, we will duplicate this vector using “RepeatVector(timeSteps)” to get a 2D array for the next layer.

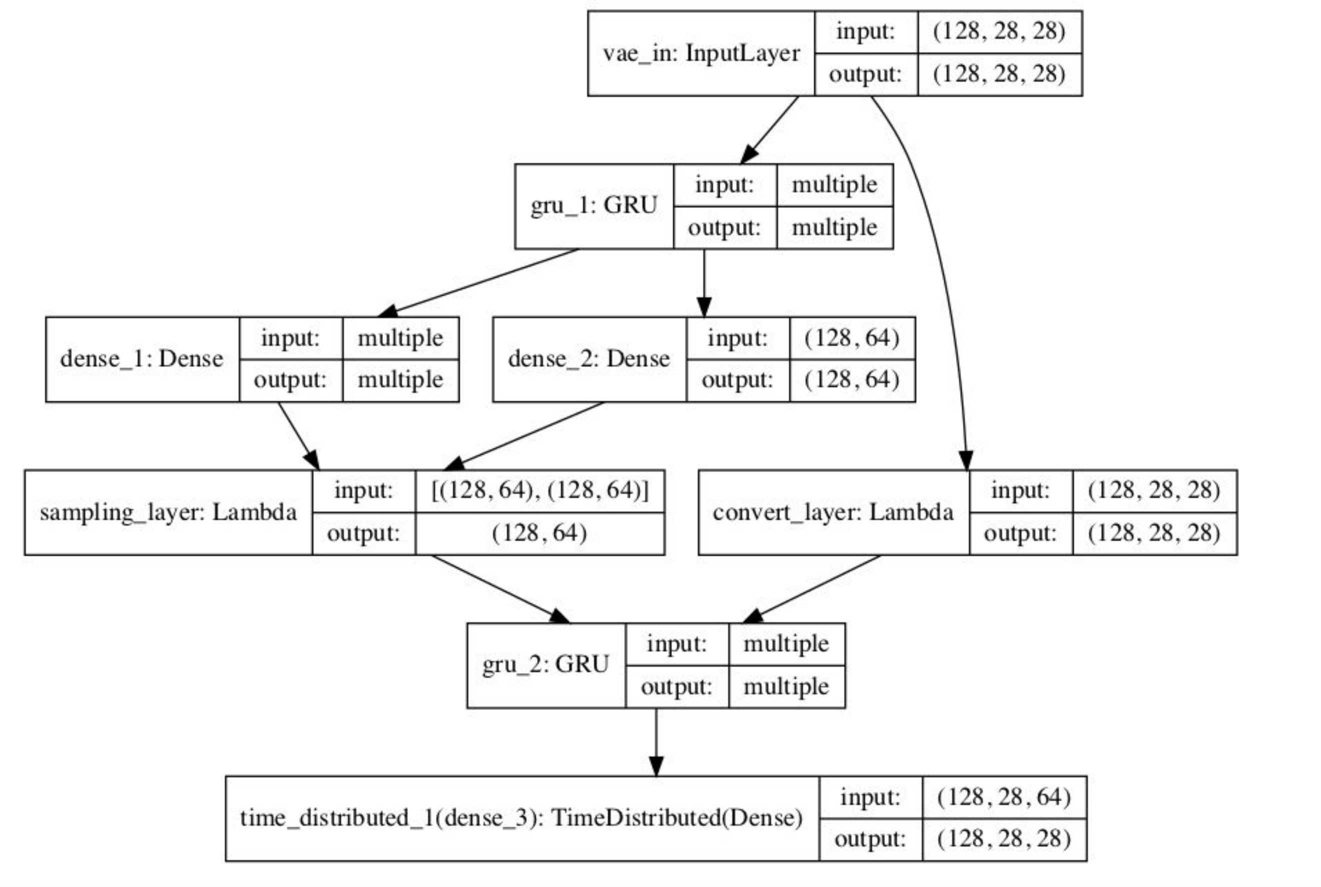
RNN architectures made no sense

Since there is no sequence in the input, they are just random consecutive numbers from 0 to 9 so it's not confidential to use the RNN for this type of problems except

1. Sorting the input in a circular way → 0, 1,2,3 .. ,8 , 9, 0, 1 2 .. and learn the natural counting pattern in order to identify whether an input sequence is in correct counting order or not.

2. Using videos instead of images and predicting the next frame to enhance the searching speed.

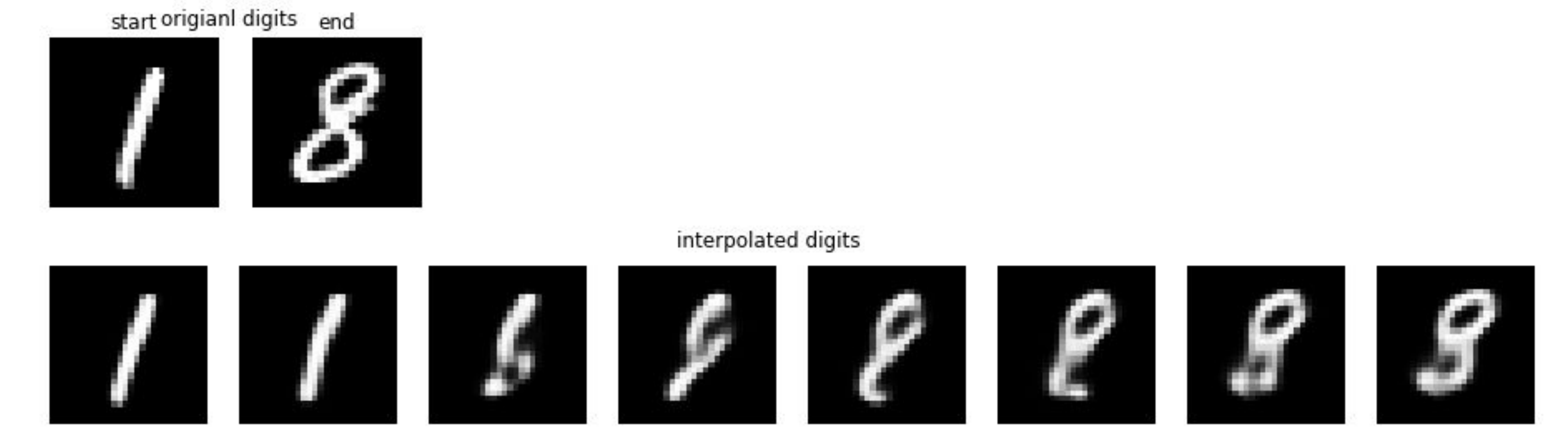
RNN (GRU) Model Architecture



67.4471

Evaluated Loss

Interpolation example (8 steps)



Our proposed solution/model and Difference from related work.

4. Finally we will try to use the **Attention model** alongside RNN Encoder-Decoder model:

A potential issue with this encoder–decoder approach is that a neural network needs to be able to compress all the necessary information of a source information into a fixed-length vector. This may make it difficult for the neural network to cope with long or big images, especially those that are larger than the images in the training corpus.

Attention model is proposed as a solution to this limitation.

Achievements

- ✓ FC layers.
- ✓ Conv + Up Sampling.
- ✓ Conv + TransposeConv.
- ✓ Conv + FC + Up Sampling.
- ✓ Conv + FC + Transpose Conv.
- ✓ Using different datasets
- ✓ Variational AutoEncoders.
- ✓ RNNs + VAE
- ■ Attention Model
- ■ Apply nearest neighbour to be done.

What we have achieved so far.

Thanks !