

# รายงานการดำเนินการสร้างภาพ

## ข้อมูลสำหรับรายงาน

ฐานข้อมูลหน้าตัวละคร : <https://www.kaggle.com/splcher/animefacedataset>

ฐานข้อมูลไฟล์ดำเนินงาน : [https://github.com/SupasanKomonlit/deep\\_learning\\_project/tree/master/generator](https://github.com/SupasanKomonlit/deep_learning_project/tree/master/generator)

ฐานข้อมูลไฟล์โมเดล : <https://drive.google.com/drive/folders/1LVzFTU-LUWzSLtyb5T8T9N8sA2F51M51?usp=sharing>

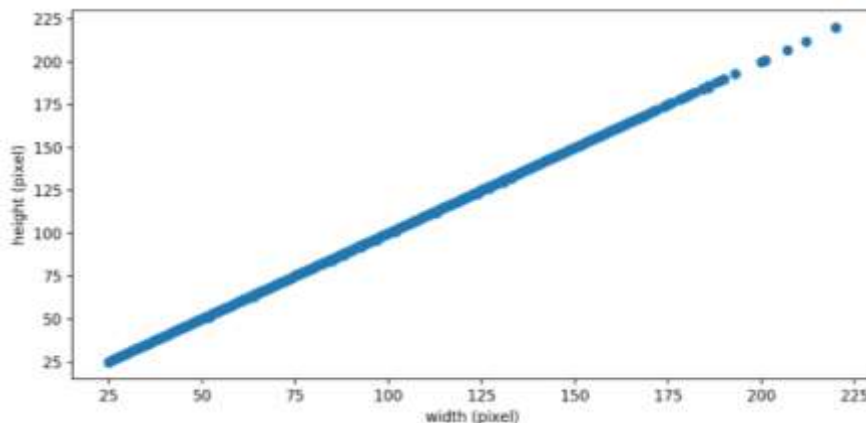
ไลบรารีในการดำเนินงาน : keras, numpy, opencv, matplotlib

## การดำเนินการทดสอบผลลัพธ์ระหว่าง VAE, GAN

### ตัวแปรของการทดสอบการดำเนินการ

การดำเนินการทดสอบความสามารถระหว่างการทำ VAE(Variational Autoencoder) กับ GAN(Generative Adversarial Network) ความพยายามที่จะกำจัดโครงข่ายให้เหมือนกัน มีดังนี้

1. ข้อมูลของขาเข้าที่เป็นรูปภาพมีขนาดความกว้างยาวที่ต่างกันดังภาพที่ 1



ภาพที่ 1 ความกว้าง ความสูงของรูปภาพในฐานข้อมูล

ก่อนที่จะนำภาพดังกล่าวมาใช้ในการดำเนินการ ผู้จัดทำดำเนินการ **Crop** ให้มีขนาดจัตุรัส แล้วดำเนินการ **resize** ให้มีขนาดเล็กที่สุดในฐานข้อมูล โดยถ้าเลขที่ได้เป็นเลขคี่จะบวก 1 เข้าไปให้เป็นเลขคู่

## 2. ขนาดของ Latent Vector

จากการดำเนินการส่วนของข้อมูลขาเข้าสำหรับการสร้างรูปภาพ ผู้จัดทำจะดำเนินการทดสอบโดยมี **Latent Vector** ขนาด 1024

## 3. Layer ในการดำเนินการ

สำหรับการดำเนินการ **Layer** ในการดำเนินการจะใช้ 3 ชั้นเสมอ โดยจะเป็นในรูปแบบ 16 32 64 คุณลักษณะ

## 4. Convolution Operation

จะมีลักษณะการดำเนินงาน 3 layer มี **kernel size** = (3,3) และมี **padding** = 'same' คือมี **padding** และสุดท้ายลำดับการ **strides** หรือการขยับ **filters** จะมีเป็น 1, 2, 1 กล่าวคือการทำงานครั้งที่ 2 จะมีขนาดลด หรือเพิ่มขึ้น 2 เท่านั้นเอง

## 5. Activation

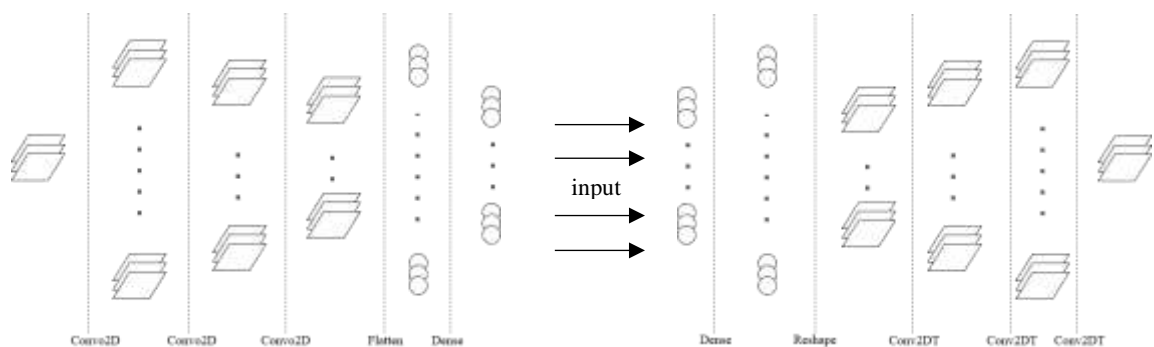
ในการดำเนินงานส่วนของ **Activation function** สำหรับการดำเนินการใน **Convolution** ทั้ง 2 โมเดลที่นำมาเทียบประสิทธิภาพจะใช้ **activation** เดียวกันคือ **relu**

## 6. รูปแบบการกระจายตัวของข้อมูล

ในการดำเนินการจะมีการใช้ค่าความน่าจะเป็นเข้ามาช่วย โดยทางผู้จัดทำจะกำหนดให้รูปแบบการกระจายตัวของข้อมูลอยู่ในแบบ **normal distribution** โดยมีค่าเฉลี่ย กับส่วนเบี่ยงเบนมาตรฐาน อยู่ที่ 0 และ 1 ตามลำดับ

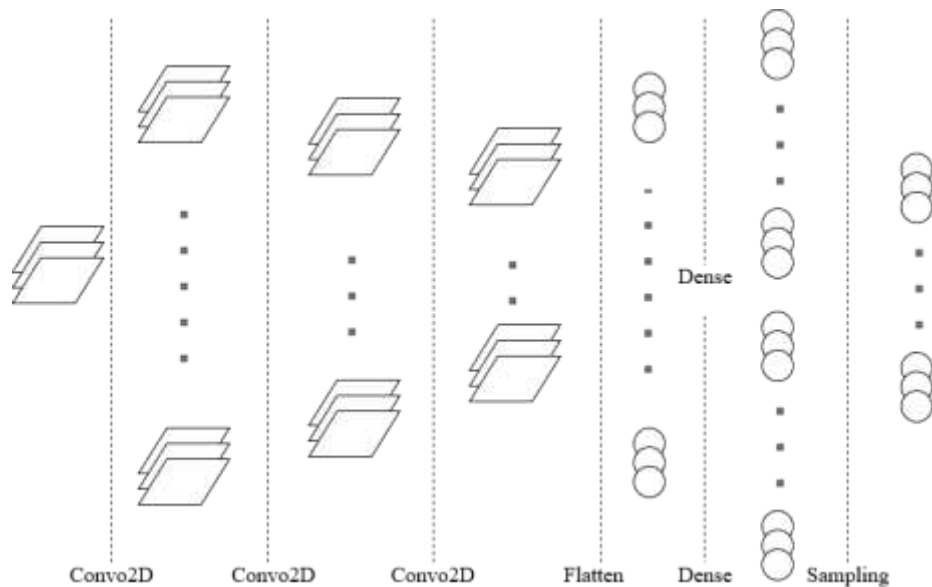
## โครงสร้างของระบบ VAE

ระบบ **VAE** จะมีโครงสร้างคล้ายกับ **Autoencoder** ในส่วนของการดำเนินการที่แล้ว มีระบบดังภาพที่ 2



ภาพที่ 2 ภาพระบบ autoencoder

สิ่งที่แตกต่างจากเดิมคือในส่วนของ **encoder** จะมีการเพิ่ม **layer** คั่นกลางก่อนที่จะออกเป็น **output** ที่เป็น **latent vector** โดยจะเปลี่ยนเป็นไปดังภาพที่ 3



ภาพที่ 3 ภาพระบบ variational encoder

จากภาพจะพบว่าหลังจาก **flatten** แล้วจะนำข้อมูลที่ได้ไปเป็นข้อมูลขาเข้า 2 ส่วนด้วยกัน แล้วจึงนำ 2 ส่วนดังกล่าวเข้าสู่ตัว **latent vector** ที่เป็น **output**

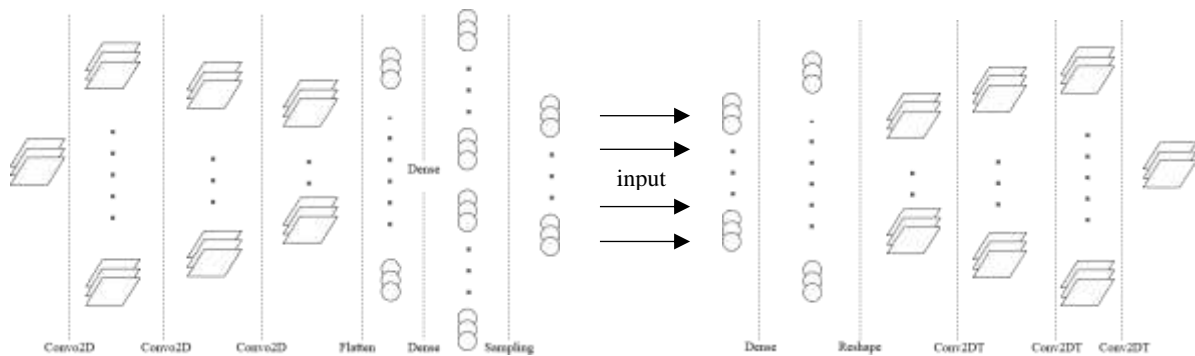
ข้อมูล 2 ส่วนที่เพิ่มเข้ามานั้นจะทำหน้าที่เป็น **mean** กับ **standard deviation** โดยจะถูกดำเนินการส่งเป็น **output** ด้วยการดำเนินการดังภาพที่ 4

```
def sampling( args ): # Function output of Variational Encoder
    mean, variance = args
    epsilon = K.random_normal( shape = K.shape( mean ), mean = _MEAN, stddev = _STDDEV )
    return mean + K.exp( variance / 2 ) * epsilon
```

ภาพที่ 4 การดำเนินการหา **latent vector output** ส่วน variational encoder

จากภาพเป็นการดำเนินการบนฟังก์ชัน Keras ตัวแปร **K** คือ **keras.backend** โดย **mean** และ **variance** คือ **layer** ทั้ง 2 ส่วนที่ถูกเพิ่มขึ้นมาขึ้นนั่นเอง

จึงสามารถสรุปภาพรวมของระบบ VAE ได้ดังภาพที่ 5

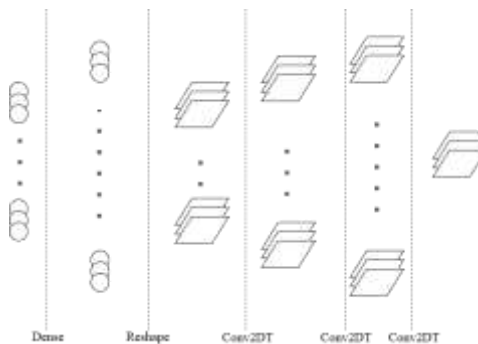


ภาพที่ 5 ภาพรวมของระบบ variational autoencoder

## โครงสร้างของระบบ GAN

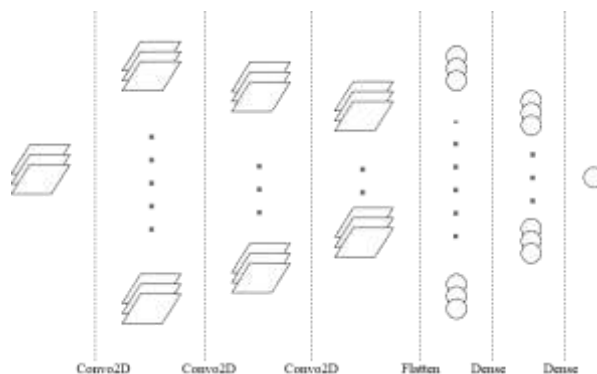
การดำเนินงานของระบบ GAN จะประกอบไปด้วย 2 ส่วนพื้นฐาน คือ generator และ discriminator

โมเดล generator จะทำหน้าที่สร้างภาพขึ้นมาจาก latent vector ที่เกิดจากการสุ่มเลขโดยใช้ normal distribution ดังภาพที่ 6



ภาพที่ 6 ภาพรวมระบบการ reconstruct สำหรับโมเดล generator

โมเดล discriminator จะทำหน้าที่ในการตรวจสอบภาพว่าภาพที่ได้รับมานั้นเป็นภาพจริง หรือ ภาพปลอมที่ถูกสร้างขึ้น ดังภาพที่ 7

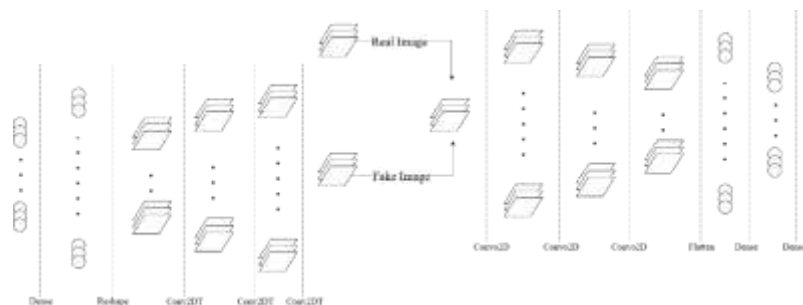


ภาพที่ 7 ภาพรวมระบบการตรวจสอบภาพโมเดล discriminator

ภาพรวมของระบบ **GAN** จะเป็นการทำงานในส่วนของทั้ง 2 ระบบทำงานร่วมกัน โดยทางตัว **Generator** จะพยายามสร้างภาพให้ผ่านการตรวจสอบ **discriminator**

## การเทรนระบบ **GAN**

การเทรนโมเดลสำหรับระบบ **GAN** จะแบ่งการเทรนออกเป็น 2 ส่วนคือการเทรน **generator** และ **discriminator** จะต้องทำควบคู่กันไป โดยภาพรวมของระบบการเทรนจะเป็นไปดังภาพที่ 8



ภาพที่ 9 ภาพรวมของการเทรนระบบ **GAN**

การเทรน **discriminator** จะเป็นการดำเนินการเทรนระบบโดยเตรียมภาพปลอม และภาพจริงใส่ระบบฝั่งขวา **discriminator** ให้สามารถทำนายได้อย่างถูกต้องว่าเป็นภาพจริง หรือเท็จ

การเทรน **generator** จะเป็นการเทรนโมเดลทางฝั่งซ้าย โดยจะปล่อยให้รันทั้งระบบดังภาพที่ 9 เพียงไม่มีการใส่ภาพจริงเข้าไป แล้วดำเนินการปรับ **weights** สำหรับฝั่งซ้ายเท่านั้น เพื่อให้ผลลัพธ์ทาง **discriminator** ทำนายว่าเป็นจริง

## การเทรนระบบ **VAE**

ในการเทรนระบบ **VAE** จะมีฟังก์ชัน **loss** ที่แตกต่างออกไปจากการคำนวณ โดยฟังก์ชันสำหรับคำนวณมีดังภาพที่ 10

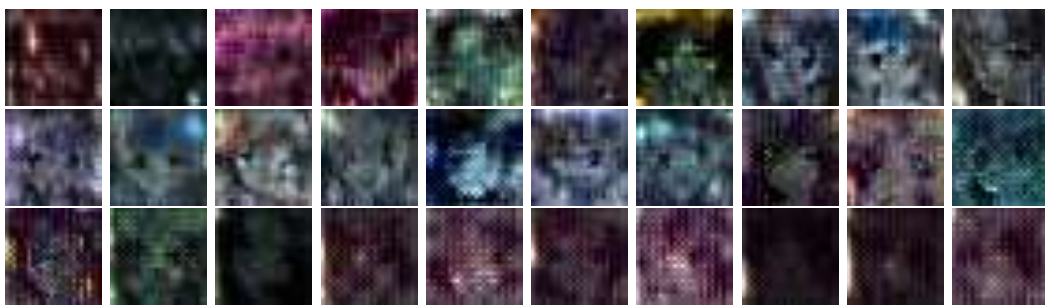
```
def kl_loss( y_true , y_pred ):  
    loss = -0.5 * K.sum( 1 + variance_layer - K.square( mean_layer ) - K.exp( variance_layer ),  
        axis = 1 )  
    return loss  
  
def r_loss( y_true , y_pred ):  
    return K.mean( K.square( y_true - y_pred ), axis = [ 1 , 2 , 3 ] )  
  
def total_loss( y_true , y_pred ):  
    return _LOSS_FACTOR * r_loss( y_true , y_pred ) + kl_loss( y_true , y_pred )
```

ภาพที่ 10 ฟังก์ชันการคำนวณค่า **loss** ของระบบ

โดยตามปกติสำหรับการดำเนินการ **autoencoder** จะมีการหาค่า **loss** โดยใช้ ค่า **mean square** ดังฟังก์ชัน **r\_loss** แต่การดำเนินการ **VAE** จะมีการใช้ **kl divergence** ดังฟังก์ชัน **kl\_loss** ที่นำค่าในส่วนของ **layer** ที่เพิ่มขึ้นมามาคำนวณด้วย

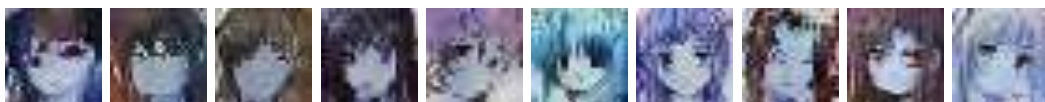
### ผลลัพธ์สำหรับการดำเนินการ **GAN**

ในการทดสอบผู้จัดทำได้ดำเนินการทดสอบในรูปแบบของการดำเนินสลับการเทรน **discriminator** และ **generator** โดยได้มีการทดสอบอยู่ 2 แบบ ได้แก่ การดำเนินการข้อมูลทีละ 2048 รูป กับ การดำเนินการข้อมูลทีละ 32 รูป เปลี่ยนเหมือนการแบ่ง **batch** สำหรับการปรับ **weights** แต่ในกรณีคือการแบ่ง **set** ข้อมูลในการเทรนโมเดล



ภาพที่ 11 ผลลัพธ์ในแต่ละรอบของการดำเนินการทีละ 2048 รูป

จากชุดภาพที่ 11 คือตัวอย่างการดำเนินการสร้างภาพ **latent vector** ที่สุ่มมา โดยทั้งหมด 30 รอบ นับจากซ้ายไปขวา

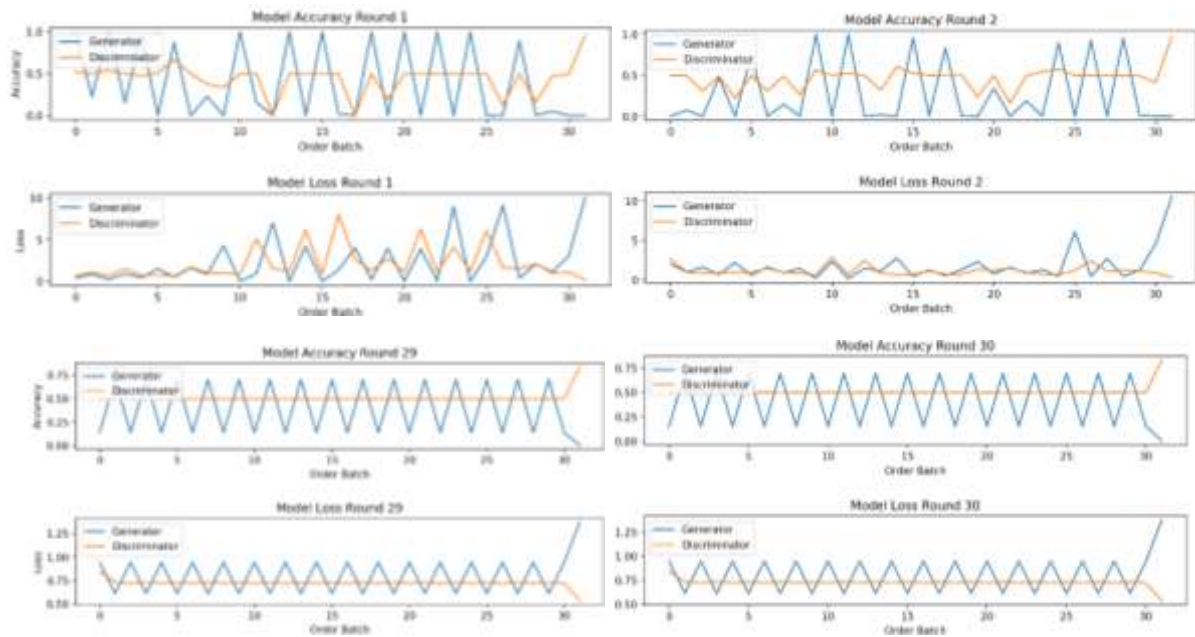


ภาพที่ 12 ผลลัพธ์ในแต่ละรอบของการดำเนินการทีละ 32 รูป

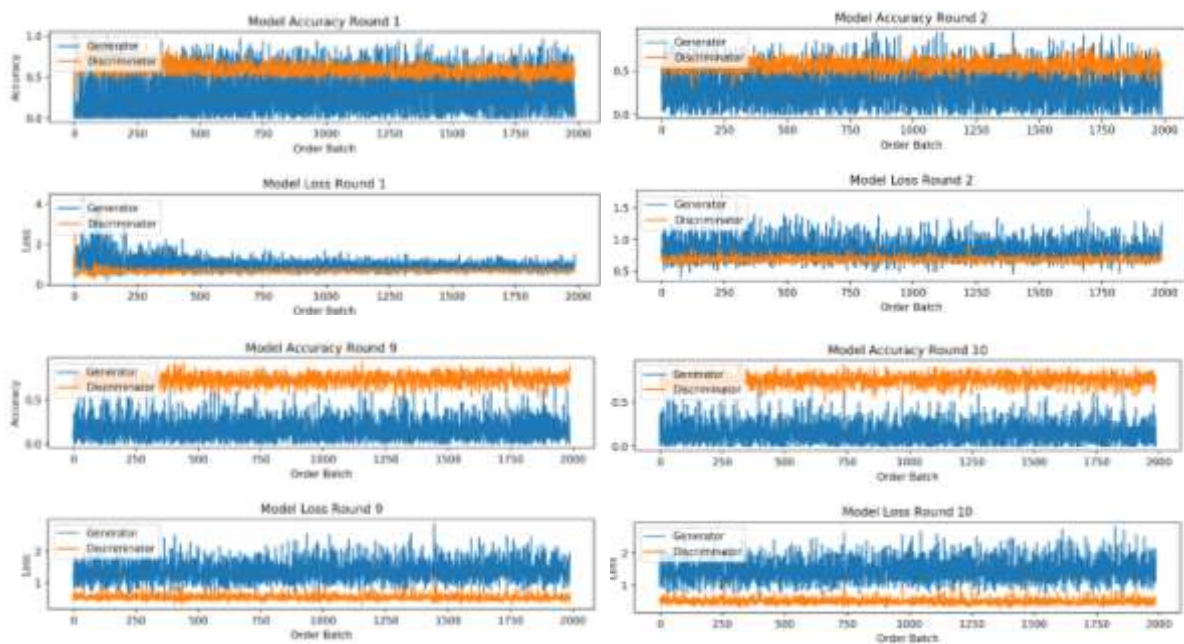
จากชุดภาพที่ 12 คือตัวอย่างการดำเนินการสร้างภาพ **latent vector** ที่สุ่มมา โดยทั้งหมด 10 รอบ นับจากซ้ายไปขวา

จากภาพที่ 11 – 12 แสดงให้เห็นถึงความแตกต่างของผลลัพธ์ที่ได้ ว่าการดำเนินการปรับทีละโมเดลทีละเยอะ ๆ ทำให้ได้ผลลัพธ์ที่ได้ออกมาอยู่แนวสีโทนเดียวกันมากกว่าการดำเนินการทีละน้อย ๆ

ในลำดับต่อมา ลองพิจารณาประวัติการเทรนในแต่ละรอบ



ภาพที่ 13 ตัวอย่างประวัติการเทรนข้อมูลทีละ 2048 รูป



ภาพที่ 14 ตัวอย่างประวัติการเทรนข้อมูลทีละ 32 รูป

จากภาพที่ 13 จะเห็นว่า 2 รอบสุดท้ายผลลัพธ์ที่ได้เริ่มที่จะเป็นไปในทิศทางเดิมจึงไม่ก่อให้เกิดการเปลี่ยนเท่าไรแล้ว จึงทำให้ระบบจบการพัฒนาไปในทางที่แตกต่างมากยิ่งขึ้น

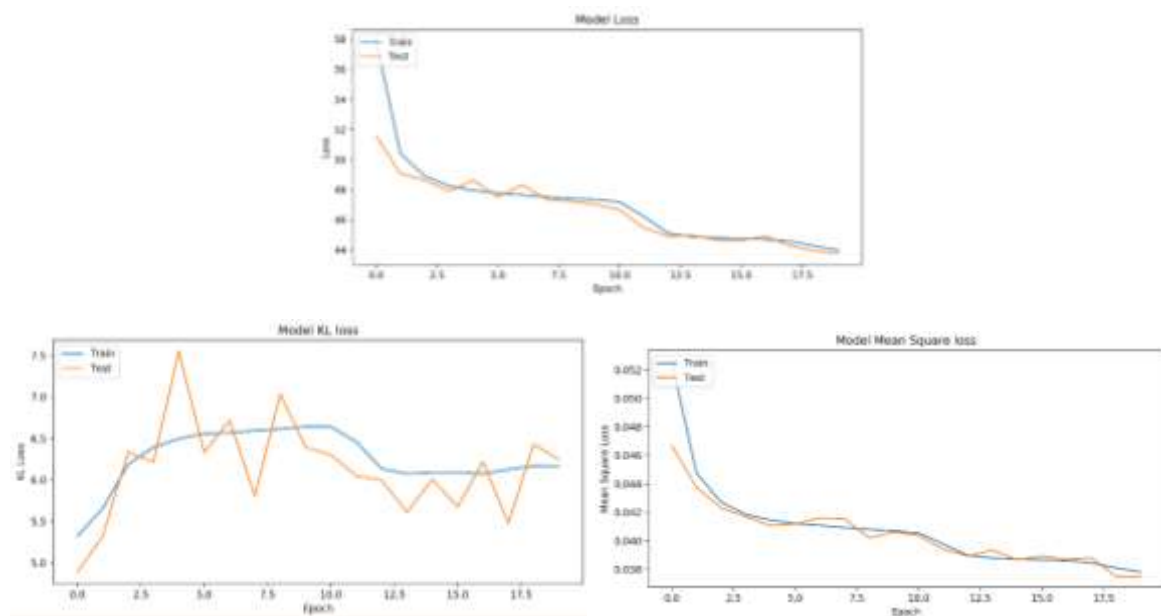
จากภาพที่ 14 จะเห็นว่าพอช่วงท้ายระบบตรวจสอบเริ่มแยกห่างจากระบบสร้างภาพ นั้นหมายความว่าระบบตรวจสอบเริ่มสามารถจับเอกลักษณ์หรือคุณลักษณะของภาพที่จริงได้แล้วนั่นเอง



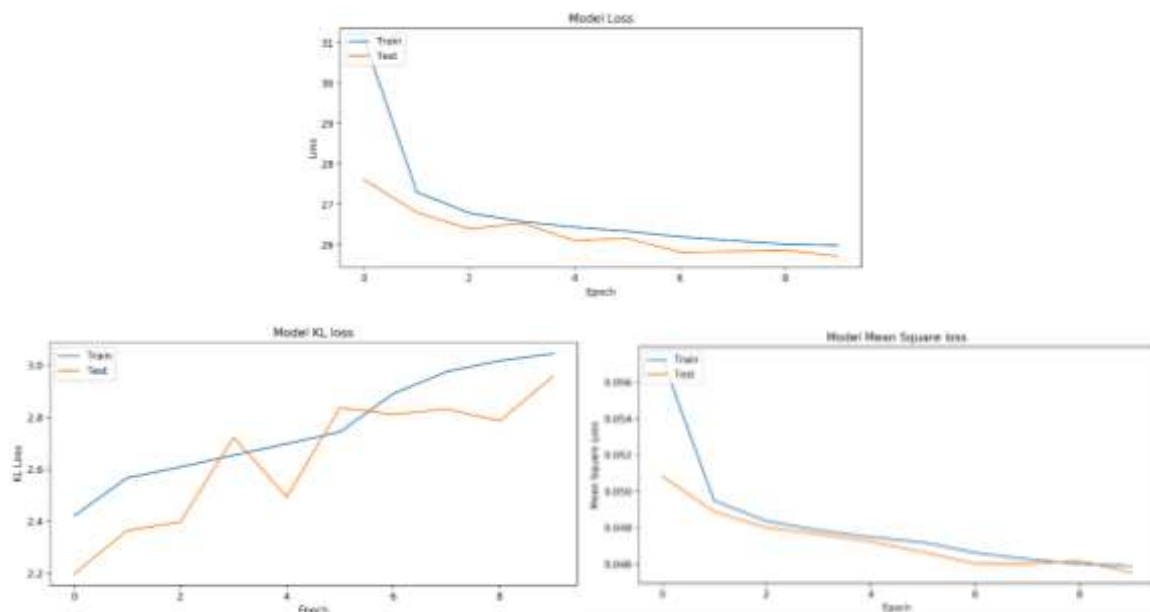
ทั้งนี้อาจเป็นเพราะการดำเนินดังกล่าวอาจไม่มีภาพปลอมสำหรับการเทรน 2048 รูปต่อครั้ง จึงทำให้ระบบการตรวจสอบภาพยังไม่สามารถทำงานได้มีประสิทธิภาพที่จะทำให้ระบบสร้างภาพ สร้างภาพดี ๆ ขึ้นมา

### ผลลัพธ์สำหรับการดำเนินการ VAE

ในการดำเนินการทดสอบผู้จัดทำได้ดำเนินการทดสอบ โดยการปรับพารามิเตอร์ `_LOSS_FACTOR` ที่ทำหน้าที่เป็นสัดส่วนของค่า `loss` ระหว่างการ `mean square` กับ `kl divergence` ดังตัวเลข 1000 กับ 500



ภาพที่ 15 ประวัติการเทรน VAE กรณี loss factor 1000



ภาพที่ 16 ประวัติการเทรน VAE กรณี loss factor 500





ภาพที่ 17 ภาพผลลัพธ์การทำงานของ VAE กรณี loss factor 1000



ภาพที่ 18 ภาพผลลัพธ์การทำงานของ VAE กรณี loss factor 500

จากการดำเนินงานพบว่ากรณีที่ **loss factor** หรือให้น้ำหนักกับภาพวิธีหา **loss** สำหรับ **autoencoder** น้อยเพียงใด ภาพจะมีความเปลี่ยนแปลงจากต้นฉบับมากขึ้นเท่านั้น แต่ก็อาจก่อให้เกิดความจำเจบางอย่างได้ ลองพิจารณาภาพที่ 18 จะเห็นว่าแปลงจากต้นฉบับได้แค่โครงหน้าที่เหมือนคล้ายเดิมตลอดเลย

**สรุปผลการดำเนินงานและวิเคราะห์ผลลัพธ์**



ภาพที่ 19 เปรียบเทียบ VAE กับ GAN

ภาพที่ 19 แสดงการเปรียบเทียบผลลัพธ์การสร้างภาพพบว่า VAE จะมีได้ผลลัพธ์ที่มีองค์ประกอบเป็นหน้าการดูน่าเชื่อถือเนื่องจากการดำเนินการที่พยายามสร้างให้เหมือนภาพต้นแบบ โดยมีการเปลี่ยนแปลงทางด้านการ **encoder** จึงทำให้คงรูปได้อย่างแม่นยำ

ส่วน GAN จะได้ภาพที่มีหลากหลายลักษณะ และสีสรรมากกว่า เนื่องจาก GAN เป็นการเรียนรู้การตรวจสอบ ไม่ใช่การเรียนรู้ที่พยายามให้ได้ภาพเท่าเดิมนั่นเอง