

รายงานโครงการ วิชา Deep learning

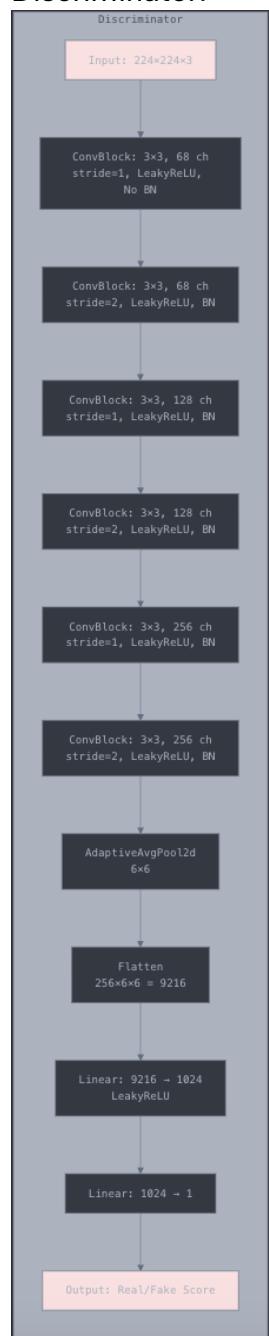
หัวข้อ :Ai ทำให้ภาพ ชัดขึ้น

ที่มาและความสำคัญ:

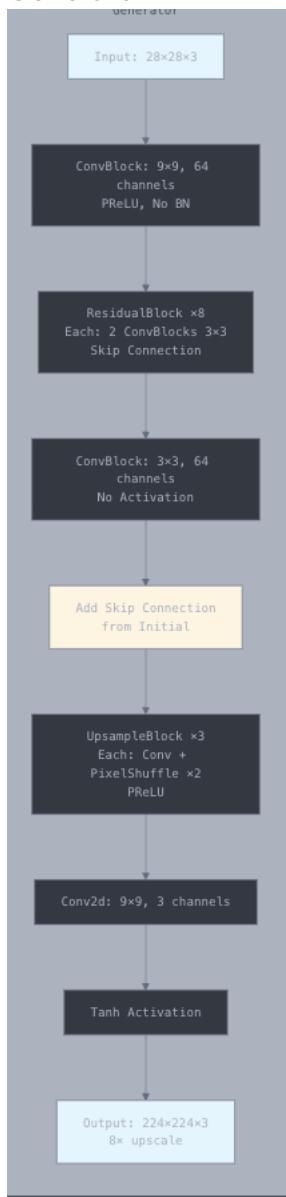
เนื่องจากโทรศัพท์ของผม มีคุณภาพของกล้องที่ไม่ได้มีคุณภาพมากนัก เลยทำให้เวลาที่เราถ่ายรูปหรือพยายาม zoom รูป ให้ไกๆ ก็จะได้รูปที่ไม่ค่อยชัดหรือ มีคุณภาพดี ซึ่งทำให้ผมได้เกิด ความคิดที่ว่า เราจะสามารถมาใช้ model deep learning ในการทำให้ภาพชัด ได้หรือไม่ โดยการ ที่เรา ที่ เรามี input เป็นรูปที่ไม่ชัดจากนั้น ให้ model ได้ไป predict ออกมาว่าเป็นอย่างไร โดยเทียบกับ รูปที่ชัด

วิธีการ: ใช้ model SRGAN ในการทำนายภาพอุปกรณ์โดย ให้ ออกแบบแต่ละ layer ดังนี้

Discriminator:



Generator:



โดย แต่ละส่วนของ code มีดังนี้

The screenshot shows a GitHub repository named "deeplearning-project". The repository is public and contains 9 commits. The files listed are .DS_Store, .gitignore, dataset.py, inference.py, main.py, model.py, and vgg_loss.py. The README file is also present. The repository has 0 stars, 0 forks, and 0 releases. There is no description or website provided.

File	Description	Last Commit
.DS_Store	use SRCNN	1 hour ago
.gitignore	fix un match dataset	6 hours ago
dataset.py	train in kaggle	1 minute ago
inference.py	use SRCNN	1 hour ago
main.py	train in kaggle	1 minute ago
model.py	train in kaggle	1 minute ago
vgg_loss.py	first commit	8 hours ago

1. main.py ซึ่งเป็น โคดที่เอาไว้ใช้ train model ซึ่งเทรนใน Kaggle ใช้ GPU Nvidia T4 / ตัว ในการ เทรน ตัวแกรนเกรน Generator อีกตัวเกรนให้ discriminator
2. dataset.py เป็น code ที่ preprocess dataset ด้วยการ อ่าน dataset จาก DIV2K จากนั้นก็เอา ไปให้ไฟล์ main.py เตรนต่อไป
3. model.py เป็น ส่วนที่สร้างตัว model ขึ้นมาซึ่งได้อธิบายไปก่อนหน้านี้แล้ว
4. vgg_loss.py เป็นส่วนที่ เอา model vgg มาช่วย คำนวณค่า loss
5. inference.py เป็นส่วนที่ลอง เอา model มาใช้จริง

อธิบายวิธีในการ train

Train discriminator:

ให้ Generator ทำให้ภาพที่รับมา ชัดขึ้น (โดยไม่คำนวณ back propagation) จากนั้น ก็ให้ discriminator ทาย จากนั้นทำซ้ำกับ label รูปที่ ชัดจากนั้น เอา ค่าที่ได้หั่งคูปิคำนวณค่า loss หารเฉลี่ย ออกมา จากนั้น หาร ด้วย gradient accumulation step เนื่องจาก ข้อมูลภาพที่ได้มีหลาย帧เลยทำให้ ไม่สามารถเอามา stack กันได้ จากนั้น ก็ปรับค่า parameters ของ discriminator เมื่อ ทำ back propagation ครบ 16 ครั้ง (ตาม gradient accumulation step)

Train generator:

ให้ generator ทำให้ภาพที่รับมาชัดขึ้น จากนั้น เอาไปให้ discriminator ทายค่าแล้วคำนวณ loss อันแรก จากนั้น ก็เอาไปเทียบ กับ ภาพชัดจริงๆ ที่ละ pixel ด้วย mean square error เป็น loss ตัวที่ 2 จากนั้น เอา model vgg มาช่วย คำนวณ L1 loss เป็น loss ตัวที่ 3
เอา หั่ง 3 มาร่วมกัน ตามสมการ

```

1      g_loss = pixel_loss + 1.0 * perc_loss + 1e-3 * adv_loss.to(0)
2      g_loss = g_loss / gradient_accumulation_step # 0
3      g_loss.backward() # 1
4
5      if (i + 1) % gradient_accumulation_step == 0:
6          opt_G.step()
7          opt_G.zero_grad()
8
9

```

แล้วก็ปรับค่า weight เมื่อกับ Discriminator

อ้างอิง:

Model: <https://arxiv.org/abs/1609.04802>

Dataset: <https://data.vision.ee.ethz.ch/cvl/DIV2K/>