Image Super-resolution using Generative Adversarial Networks

PG - Independent Study

Ankit Pant
2018201035
M.Tech. Computer Science & Engineering
International Institute of Information Technology, Hyderabad
Supervised by: Dr. Pawan Kumar

Outline

1. Introduction
2. Literature Review
3. Methodology
4. Experimentation
5. Results
6. Conclusion
7. Drawbacks
8. Future Scope
9. References

Introduction

Introduction

- Artificial Intelligence (AI) and Machine Learning being used for a large number of applications
- Generative Adversarial Networks among various models used in AI and Machine Learning (ML)
- Image Super-resolution is one such application
- PG-Independent Study aims to cover GANs and Image Super-resolution (in moderate detail)
- Study includes creating ML model and Web App to use the model

Literature Review

Generative Adversarial Networks (GANs)

- Generative Adversarial Networks (GANs) – combination of two models: generator and discriminator model working in tandem
- Generator model:
  - It is responsible for generating images (usually from noise)
- Discriminator model:
  - It is responsible to determine whether the image was originally available or generated by the generator
- Various Types of GANs include:
  - Deep Convolutional GANs (DCGAN)
  - Wasserstein GAN (WGAN)
  - Softmax GAN

Sample Training Phrase of a GAN

- The generator tries to recreate original image from noise
- This image is then input to discriminator which tries to identify whether it was generated
- The generator then again improves on the generated images
- The discriminator again determines whether the images was generated or not
- This process is repeated until the discriminator can no longer determine whether the image was generated or not $P(generated) = P(original) = 0.50$
- Both the generator and discriminator may be pre-trained to improve performance

Image Super-resolution

- **Image Super-resolution:** Conversion of (one or more) low resolution images into a high resolution image
- Benefits of increasing image resolution:
    - The resultant image is larger
    - It provides more details
    - Can be used to improve image quality as well as video quality
- Application of image super-resolution
    - Enhancing photographs and self-portraits of people
    - Enhancing surveillance footage
    - Enhancing medical diagnostic images
    - Enhancing astronomical and remotes sensing images
    - Enhancing low resolution videos

Image Super-resolution - Technique

- **Single-frame Super-resolution**
  - Traditional resolution enhancement - includes smoothing, interpolation and sharpening
  - Estimates detail that is not present
  - Training-set used to learn details of images at low resolution
  - These learned relationships used to predict details of other images
- **Multi-frame Super-resolution**
  - Works if multiple low resolution images are available of the same scene
  - Each image is naturally shifted with sub-pixel precision
  - Works when each of the images have different sub-pixel shifts

Methodology

Implementing Single-frame Super-resolution

- Approach consists of two steps:
  1. Creating and training ML model
  2. Creating the Web application to use the model
- The code is in the form of Jupyter Notebook
- Google Colab – the platform to train and test the model
- ML model currently can scale low-resolution images to a resolution of 512 x 512 pixels
- Measures to Compare Images:
  - Multi Scale Structural Similarity (MS-SSIM)
  - Mean Squared Error (MSE)
  - Peak Signal to Noise Ratio (PSNR)
  - L1 distance

Overview of the algorithm

1. Convert high resolution images to low resolution images
2. Train generator model this dataset. Store resultant generated images
3. Train discriminator model on generated and original images
4. Combine generator and discriminator in GAN and train it
5. Save the model as well as generated images.
6. Use model to upsample images to size 512 x 512
7. Run the model on test dataset
8. Import the saved models to the web application

Architecture of ML model

- Generator:
    - UNet model
    - ResNet34 as base architecture
    - Mean Squared Error (MSE) as loss function
- Discriminator:
    - basic gan_critic model
    - Cross Entropy as loss function
- Up-sampler:
    - UNet model
    - VGG16 as base architecture
    - Mean Squared Error (MSE) as loss function

Architecture of ML model

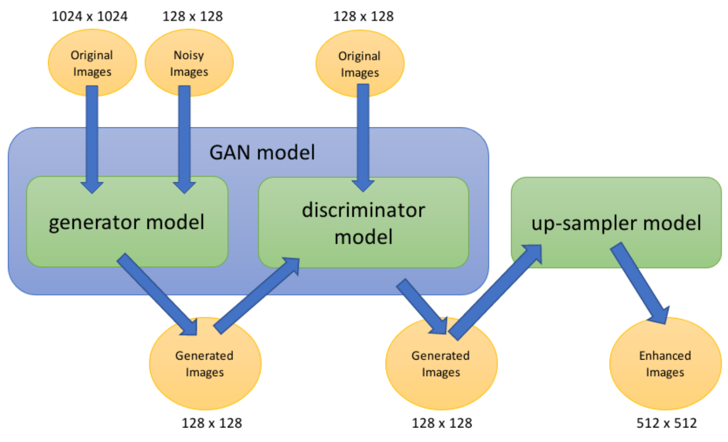

Figure 1: Architecture of ML model

Training and Test Datasets

- Training dataset (high quality):
  - 1600 images
  - Flickr-Faces-HQ Dataset
  - Created originally as a benchmark for GANs
- Test dataset (low quality):
  - 300 images
  - Selfie Data Set

Creating Web Application

- The Web Application has been created in Python
- Deployed using Docker
- Allows user to pick low resolution images
- enhances them to 512 x 512 resolution

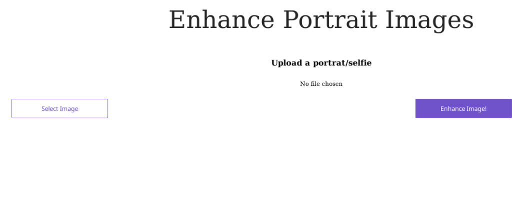

Figure 2: Web application UI (no image chosen)

Creating Web Application

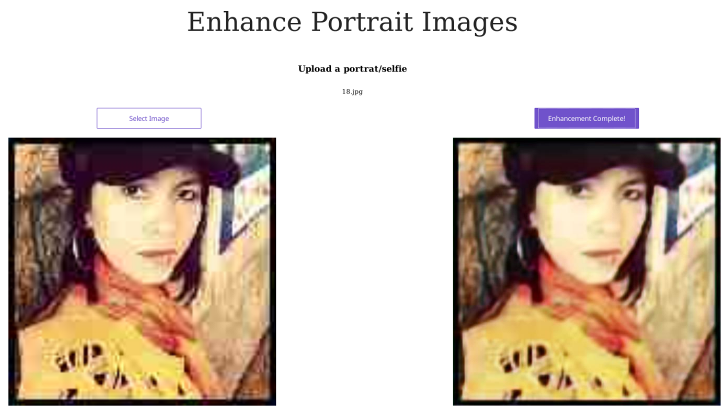

Figure 3: Web application UI (after enhancing image)

Experimentation

Experimental Setup

- Training Environment - Google Colab:
    - RAM: 25.51 GB
    - VRAM: 12 GB
    - Disk Space: 358.27 GB
- Approximate time required to train various components:
    - Generator model: 13 minutes
    - Discriminator model: 18 minutes
    - GAN: 48 minutes
    - Up-sampler: 135 minutes

Results

Comparison: Low-res images vs generated images

- Both low res images and generated images compared to original images

Table 1: Comparison between low-res and initially generated images

	Image type	average SSIM	average MSE	average PSNR	average L1
1	Low-res Image	0.127138	25652.10	4.2958	3.603e+06
2	Generated Image	0.135904	25757.22	4.3048	3.611e+06

Comparison: Low-res images vs enhanced images

- Both low res images and generated images compared to original images

Table 2: Comparison between low-res and enhanced images

	Image type	average SSIM	average MSE	average PSNR	average L1
1	Noisy Image	0.308175	25258.53	4.3702	5.720e+07
2	Enhanced Image	0.340402	24976.29	4.4560	5.675e+07

Sample Results using VGG up-sampler

Figure 4: Results of image enhancement (vgg)

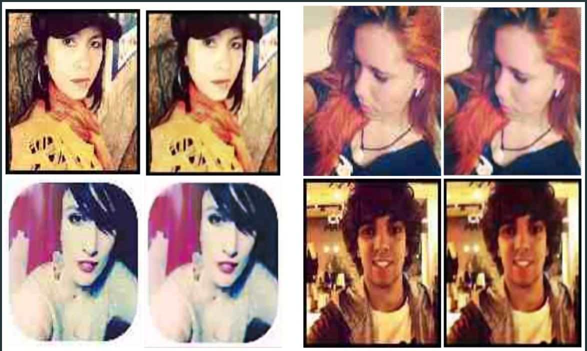

Figure 5: Results of image enhancement (web-app) (zoom in to view difference)

Conclusion

Conclusion

- AI and ML models getting better – can even outperform humans
- This Independent Study applies developments in ML to enhance low resolution images
- ML models created that can be imported by other applications
- ML models created that can be used for transfer learning
- Created Web application that can perform image enhancements
- Can have commercial application as well

Drawbacks

Drawbacks of current implementation

- Maximum output resolution of 512 x 512
- Works only for low resolution images
- Web application unable to use VGG based up-sampler
- Model trained on relatively small dataset
- Can have extra smoothing effect
- Output of images containing certain scenes overexposed

Future Scope

Future Scope

- Current Implementation
  - Output resolution may be increased
  - Training loop can be extended for better predictions
  - Size of training dataset can be increased
  - More complex underlying architectures can be used
  - Web application can be deployed on cloud (e.g. AWS)
- Other Approaches:
  - Using perceptual loss functions
  - Using Automated Texture Analysis
  - Incremental training
    1. Train 64x64 images, followed by
    2. Train 128x128 images, followed by
    3. Train 256x256 images, etc.

References

- Goodfellow, Ian J. and Pouget-Abadie, Jean and Mirza, Mehdi and Xu, Bing and Warde-Farley, David and Ozair, Sherjil and Courville, Aaron and Bengio, Yoshua, “Generative Adversarial Nets”, arXiv e-prints, https://arxiv.org/pdf/1406.2661.pdf
- “Generative Adversarial Network”, Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Generative_adversarial_network

- Alec Radford, Luke Metz, Soumith Chintala, “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, arXiv e-prints, https://arxiv.org/pdf/1511.06434.pdf
- Martin Arjovsky, Soumith Chintala, Léon Bottou, “Wasserstein GAN”, arXiv e-prints, https://arxiv.org/pdf/1701.07875.pdf
- Min Lin, “Softmax GAN”, arXiv e-prints, https://arxiv.org/pdf/1704.06191.pdf
- Kevin Su, “Introduction to Image Super-resolution”, http://www.cs.utsa.edu/~qitian/seminar/Fall04/superresolution/SR_slides_xsu.pdf

References iii

- Jeremy Howard et al, “fastai”, https://docs.fast.ai/, https://www.fast.ai/
- NVlabs, “Flickr-Faces-HQ Dataset (FFHQ)”, https://github.com/NVlabs/ffhq-dataset
- Center for Research in Computer Vision, “Selfie Data Set”, University of Central Florida https://www.crcv.ucf.edu/data/Selfie/
- Olaf Ronneberger, Philipp Fischer, Thomas Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, arXiv e-prints, https://arxiv.org/pdf/1505.04597.pdf

- Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, arXiv e-prints, https://arxiv.org/pdf/1512.03385.pdf
- “Mean squared error”, Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Mean_squared_error
- “Cross entropy”, Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Cross_entropy
- Neurohive, “VGG16 – Convolutional Network for Classification and Detection”, https://neurohive.io/en/popular-networks/vgg16/

References v

- “Structural similarity”, Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Structural_similarity#Multi-Scale_SSIM
- “Peak signal-to-noise ratio”, Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Peak_signal-to-noise_ratio
- “Least absolute deviations”, Wikipedia, the free encyclopedia, https://en.wikipedia.org/wiki/Least_absolute_deviations
- “L1 norm”, Wikipedia, the free encyclopedia, https://en.wikipedia.org/w/index.php?title=L1_norm&redirect=no

- Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi, “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network”, arXiv e-prints, `https://arxiv.org/pdf/1609.04802.pdf`
- Mehdi S. M. Sajjadi, Bernhard Schölkopf, Michael Hirsch, “EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis”, arXiv e-prints, `https://arxiv.org/pdf/1612.07919.pdf`