

Compresión de imágenes sin pérdida a través de la súper resolución

[Sheng Cao](#) , [Chao-Yuan Wu](#) , [Philipp Krähenbühl](#) .

Citación

```
@article { cao2020lossless ,  
  title = { Compresión de imagen sin pérdida a través de la súper resolución  
} ,  
  author = { Cao, Sheng y Wu, Chao-Yuan y Kr { \ "a } henb { \ " u } hl, Philipp  
} ,  
  año = { 2020 } ,  
  journal = { arXiv preprint arXiv: 2004.02872 } ,  
}
```

Si usa nuestra base de código, considere también [citar L3C](#)

Visión general

Esta es la implementación oficial de SReC en [PyTorch](#) . SReC enmarca la compresión sin pérdidas como un problema de súper resolución y aplica redes neuronales para comprimir imágenes. SReC puede lograr tasas de compresión de vanguardia en grandes conjuntos de datos con tiempos de ejecución prácticos. El entrenamiento, la compresión y la descompresión son totalmente compatibles y de código abierto.

Empezando

Recomendamos los siguientes pasos para comenzar.

1. [Instala las dependencias necesarias](#)
2. [Descargue el conjunto de validación de Open Images](#)
3. [Ejecute la compresión en el conjunto de validación de imágenes abiertas](#) con [pesos de modelos entrenados](#)

Instalación

Consulte [aquí las](#) instrucciones de instalación.

Pesas modelo

Hemos lanzado modelos entrenados para [ImageNet64](#) y [Open Images \(PNG\)](#) . Todos los resultados de compresión se miden en bits por subpíxel (bpsp).

Conjunto de datos	Bpsp	Pesas modelo
ImageNet64	4.29	models / imagenet64.pth
Imágenes abiertas	2,70	modelos / openimages.pth

Formación

Para ejecutar el código, debe estar en el directorio de nivel superior.

```
python3 -um src.train \
  --train-path "path to directory of training images" \
  --train-file "list of filenames of training images, one filename per line" \
  \
  --eval-path "path to directory of eval images" \
  --eval-file "list of filenames of eval images, one filename per line" \
  --plot "directory to store model output" \
  --batch "batch size"
```

Las imágenes de entrenamiento deben organizarse en forma de train-path/filenamenombre de archivo en el archivo de tren. Lo mismo se aplica a las imágenes eval.

Hemos incluido nuestros archivos de capacitación y evaluación utilizados para ImageNet64 y Open Images (PNG) en el datasetsdirectorio.

Para ImageNet64, utilizamos un conjunto ligeramente diferente de hiperparámetros que los hiperparámetros de Open Images, que son los predeterminados. Para entrenar a ImageNet64 según la configuración de nuestro artículo, ejecute

```
python3 -um src.train \
  --train-path "path to directory of training images" \
  --train-file "list of filenames of training images, one filename per line" \
  \
  --eval-path "path to directory of eval images" \
  --eval-file "list of filenames of eval images, one filename per line" \
  --plot "directory to store model output" \
  --batch "batch size" \
  --epochs 10 \
  --lr-epochs 1 \
  --crop 64
```

Ejecute `python3 -um src.train --help` para obtener una lista de hiperparámetros ajustables.

Evaluación

Dado un punto de control del modelo, esto evalúa bits teóricos / subpíxel (bpsp) en función de la probabilidad de registro. El log-verosimilitud bpsp limita el bpsp de compresión real.

```
python3 -um src.eval \
  --path "path to directory of images" \
  --file "list of filenames of images, one filename per line" \
  --load "path to model weights"
```

Compresión / Descompresión

Con torchac instalado, puede ejecutar compresión / descompresión para convertir cualquier imagen en archivos .srec. Lo siguiente comprime un directorio de imágenes.

```
python3 -um src.encode \
  --path "path to directory of images" \
  --file "list of filenames of images, one filename per line" \
  --save-path "directory to save new .srec files" \
  --load "path to model weights"
```

Si desea un tiempo de ejecución preciso, le recomendamos ejecutar Python con `-obandera` para deshabilitar las afirmaciones. También incluimos un `--decode` indicador opcional para que pueda verificar si descomprimir el archivo .srec proporciona la imagen original, así como proporcionar tiempo de ejecución para la decodificación.

Para convertir archivos .srec a PNG, puede ejecutar

```
python3 -um src.decode \
  --path "path to directory of .srec images" \
  --file "list of filenames of .srec images, one filename per line" \
  --save-path "directory to save png files" \
  --load "path to model weights"
```

Descargando ImageNet64

Puede descargar los conjuntos de capacitación y validación de ImageNet64 [aquí](#).

Preparación de un conjunto de datos de imágenes abiertas (PNG)

Utilizamos el mismo conjunto de imágenes de capacitación y validación de Open Images que [L3C](#) .

Para **imágenes de validación** , puede [descargarlas aquí](#) .

Para **imágenes de entrenamiento** , por favor clone el [repositorio L3C](#) y ejecute el [script desde aquí](#)

Consulte [este problema](#) para conocer las diferencias entre Open Images JPEG y Open Images PNG.

Reconocimiento

Gracias a [L3C](#) por implementaciones de EDSR, mezclas logísticas y codificación aritmética. Un agradecimiento especial a [Fabian Mentzer](#) por informarnos sobre problemas con el script de preprocesamiento para Open Images JPEG y resolverlos rápidamente.