Lossy Compression

Can you tell a difference?

Can you tell a difference?

363 kB

110 kB !!

Lossy Compression

Last class, we discussed **Lossless** compression, in which the data can be restored to its original state when it's **decompressed**.

Lossy compression is what happens when we throw away data in the compression process in order to save space!

In many forms of digital media, we can lose a lot of data and not have the result be noticeably different from the original!

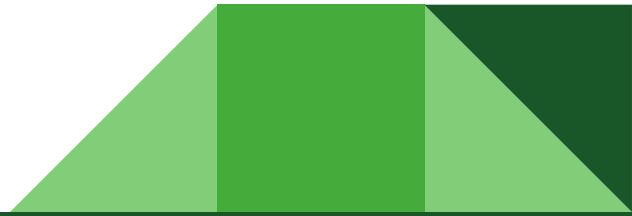
Lossy Compression

Pros

- Compressed version is **significantly** smaller than the original
  - Lossy Compression can achieve much smaller relative sizes than Lossless Compression
- Often not even noticeable by human eyes/ears

Cons

- The data, when decompressed, is no longer identical to the original
  - It's very close though!



Lossy Compression

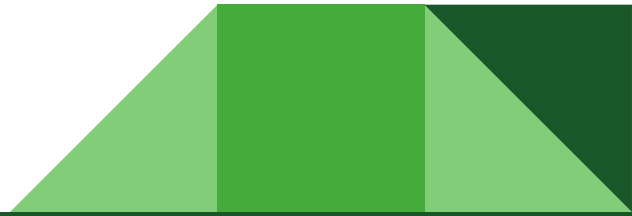
Algorithms that perform Lossy Compression are significantly more common than ones that perform Lossless Compression, because the size savings Lossy Compression offers.

Here are some common Lossy Compression algorithms:

JPEG is used for **images**

MP3 is used for **music**

MPEG-4 is used for **video**



Lossy Compression

Algorithms that perform Lossy Compression are significantly more common than ones that perform Lossless Compression, because the size savings Lossy Compression offers.

Here are some common Lossy Compression algorithms:

JPEG is used for **images** - .jpg/jpeg

MP3 is used for **music** - .mp3

MPEG-4 is used for **video** - .mp4

Like we discussed last class, the file extension used indicates what kind of compression is being used!

Lossy Compression

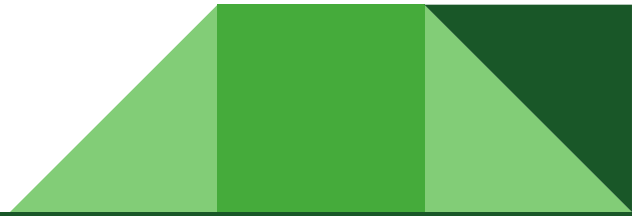
Lossy Compression algorithms are most commonly used on **images**, **audio**, and **video**. This is because there are elements within each of those that our human senses will not notice if they're missing.

Lossy Compression is much less common with text, because it's much easier to notice when there's data missing in text!

Example: Remove Vowels

One possible Lossy Compression algorithm for text is that of removing the vowels.

Cn y rd wht ths txt sys wtht vwls?



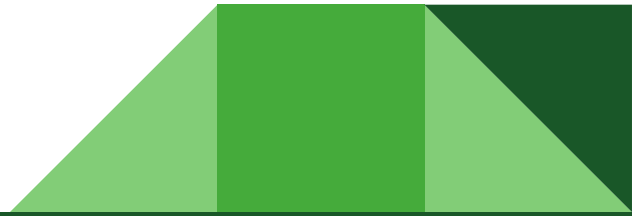
xmpl: Rmv Vwls

One possible Lossy Compression algorithm for text is that of removing the vowels.

Cn y rd wht ths txt sys wtht vwls?

Can you read what this text says without vowels?

Using an algorithm like this can sometimes be useful - humans are pretty good at interpreting words, even without vowels, so if we really need to save some space, this can be an effective solution!



Lossy Compression with Images

One way to perform lossy compression with images is to decrease the resolution.

When an image's resolution is lowered, we're representing the same image using fewer pixels.

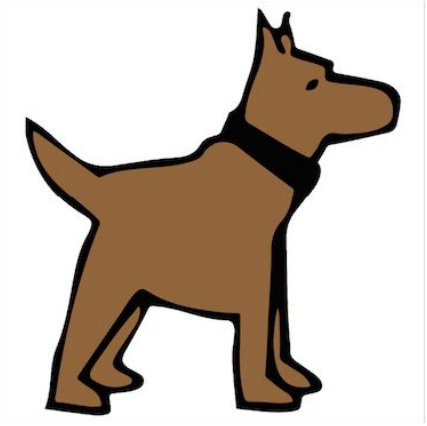
363 kB

110 kB

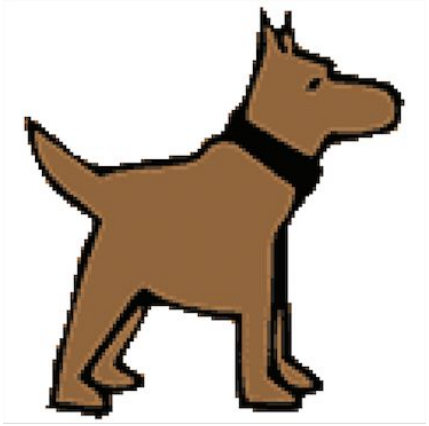
Lossy Compression with Images

One way to perform lossy compression with images is to decrease the resolution.

When an image's resolution is lowered, we're representing the same image using fewer pixels.



637 x 637 pixels



90 x 90 pixels

All Digital Data is Compressed

At a very basic level, all digital data is technically a compressed form of the information it represents.

This is because we live in an **analog** world, and computers are only capable of storing **digital** data. Digital data can only be comprised of 1's and 0's, on and off. The real world is full of all kinds of in-betweens, though!

Space is the biggest limiting factor in approximating analog data with digital data. The closer you get to representing analog data, the more space that's going to consume!