

<https://readcoop.eu>

ARTIFICIAL INTELLIGENCE

Can AI save bad scans?



2 years ago

Felix Dietrich



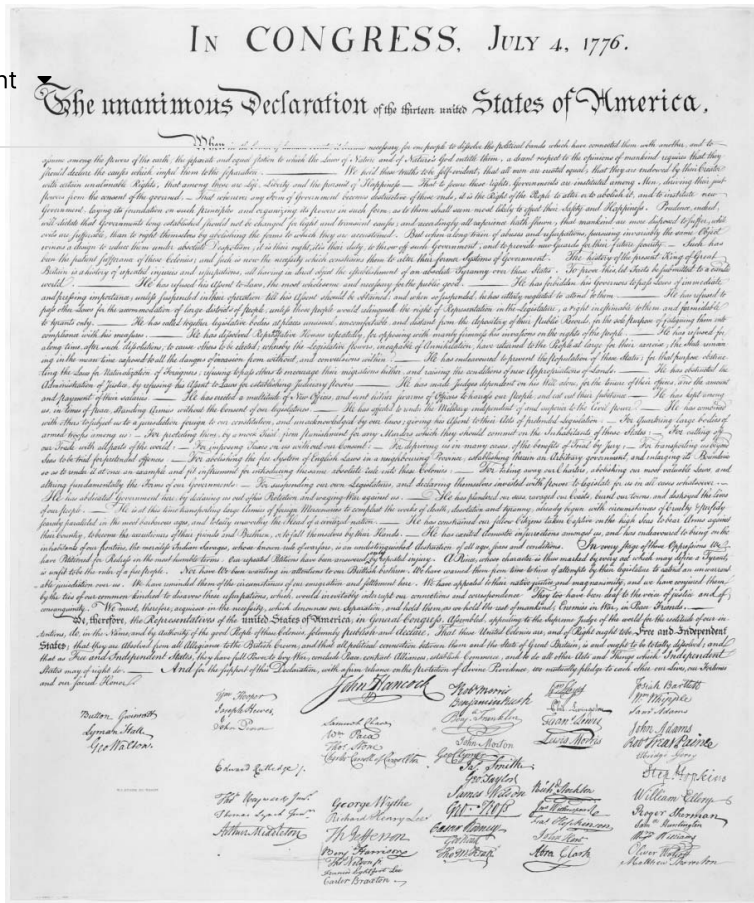
The starting point for any kind of document digitization, whether done by hand or through sophisticated text recognition algorithms, is a good-quality image. Take a look at the one below. It is a scan of the US declaration of independence – but not of the original. The real one has suffered badly (<https://www.ushistory.org/declaration/document/scan.html>) due to improper storage and remains pretty washed out as of today. The one below is a facsimile created by William Stone in 1823, and it has become the most commonly used copy of the declaration. It is actually a mystery (<https://www.archives.gov/publications/prologue/2003/fall/stone-engraving.html>) how Stone managed to create such a precise clone of the original parchment, but thanks to him we still have an easily readable version of this historic document.





Account

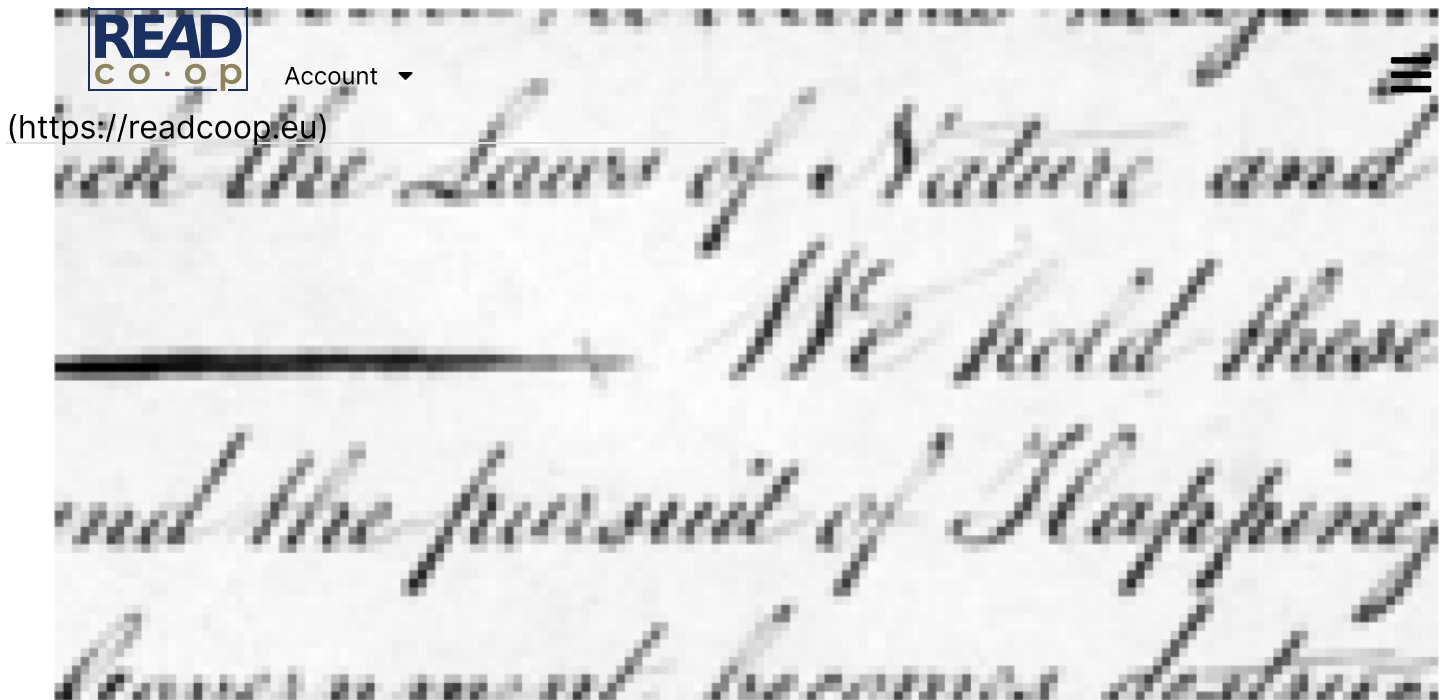
(<https://readcoop.eu>)



Source: Wikipedia

(https://commons.wikimedia.org/wiki/File:United_States_Declaration_of_Independence.jpg)

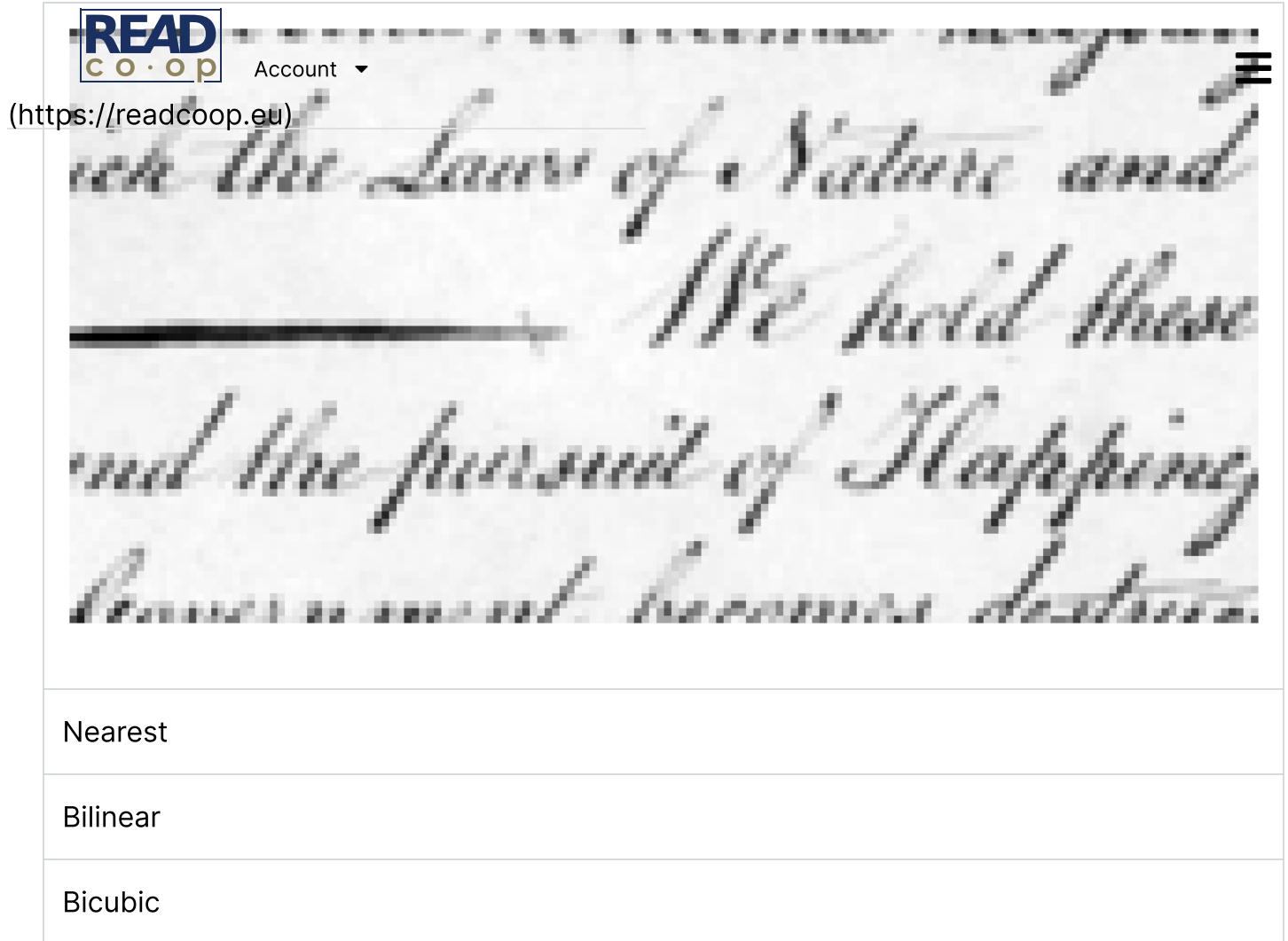
Below is a small, low resolution section of the main text. A human could still identify most letters thanks to context, but it would be a tedious task for an unfamiliar text and we can imagine that HTR algorithms won't be too happy, either, with this kind of input once the resolution gets too low. This raises a few questions: What if the original paper is lost or degraded and all that remains is a bad-quality digital scan? Or what if one has already scanned ten thousand pages, only to find that text on some of them is so small that the resolution is no longer sufficient? Do we have to scan everything again and increase our already stretched storage budgets? Maybe not.



There are several classical techniques to improve such a pixelated mess. The basic task is always to add more pixels in between existing pixels, but the question is how to choose these new pixels. The nearest-neighbour method simply takes the closest original pixel and copies it. Bilinear interpolation computes the change between neighbouring pixels and then selects an appropriate intermediate value given the position of the new pixel. Bicubic interpolation takes this up another notch by using a nonlinear function to guess an appropriate value. Alas, all of these methods suffer from a fundamental shortcoming: They can not *add* new information to an image. Where a human might be able to imagine a sharp line or a closed loop thanks to the surrounding context, these classical techniques only follow comparatively simple rules. This is where artificial neural networks can come in handy.

Compare the interpolation techniques yourself:

Low-Res Input



Last year, NVIDIA released an updated version of their deep learning supersampling algorithm, or DLSS for short. It turns out that deep learning models are now so good at improving images, that they can be used to improve the performance of real time applications. Rendering frames at lower resolution and then running them through a neural network turns out to be faster than rendering them at high resolutions in the first place, while it results in almost no perceivable image quality reduction.

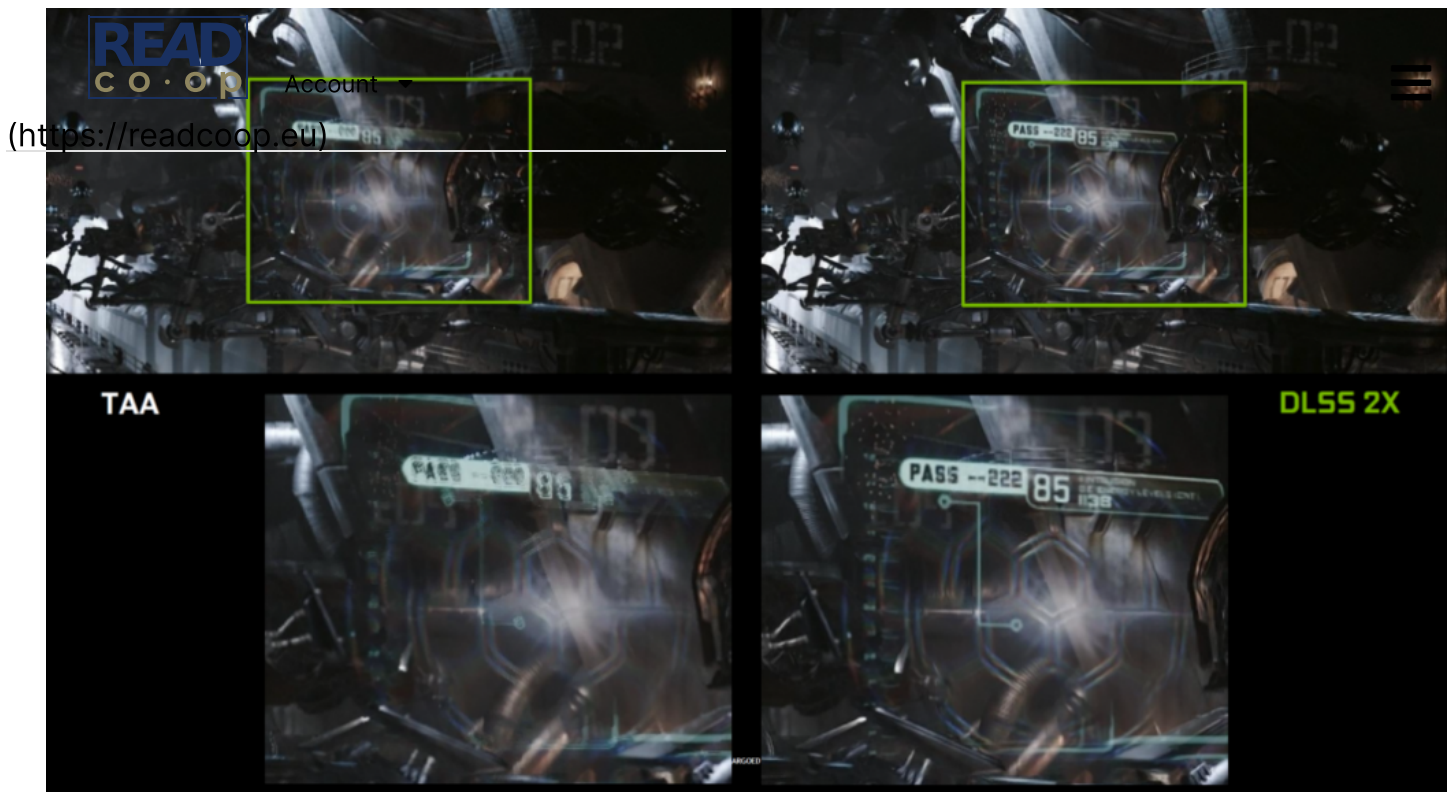


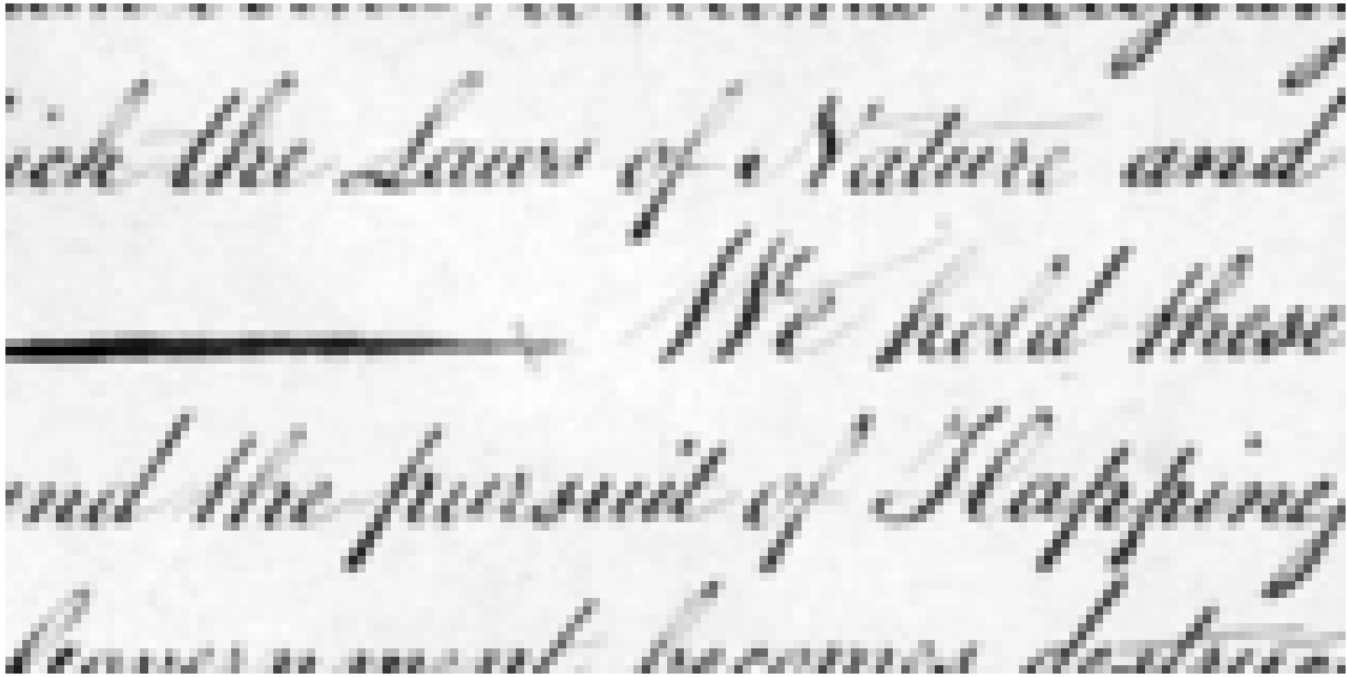
Image by NVIDIA (<https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/>)

Unfortunately, the process of upscaling real time computer graphics has certain advantages. For example, one usually has several images in a sequence that can be used to extract additional information that may be lost in individual images. One can also use additional information provided by the rendering engine, like motion vectors or even object stencils. When dealing with scanned pages of old documents, we have none of these things. We only have one image, and we have to “imagine” any kind of extra information. Fortunately, this is an area where AI has excelled (<https://deepdreamgenerator.com/>) as well. This particular sub field has made use of so called *Generative Adversarial Networks*, and while they are not yet really used in production level environments, they do show remarkable potential. They work by employing two separate neural networks: A generator and a discriminator. In the most common use case, the generator creates new images, while the discriminator tries to spot fake images among real ones from a given training dataset. The training process is a zero sum game where one network gets better at faking images while the other one gets better at identifying fakes. When trained long enough, GANs have been shown to produce photorealistic results (<https://thispersondoesnotexist.com/>). If we would like to create completely new images, we would essentially feed random data to the generator as inputs. This is very interesting for artists or content creators, but we actually want to improve

READCOOP

Account

existing images. To do that, we need a slightly modified setup, for which we took a closer look at the architecture described in this paper: *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network* (<https://arxiv.org/pdf/1609.04802.pdf>). The details are a bit too involved for this post, but the results speak for themselves.

Low-Res Input

Bicubic
AI upscale
High-Res Original

One particularly interesting feature of this model is that it was never trained on handwritten text. It was trained on the DIV2k dataset (<https://people.ee.ethz.ch/~timofter/publications/Agustsson-CVPRW-2017.pdf>), which contains a wide variation of high resolution color images showing all sorts of objects and sceneries – but no textual images.

We expect that in the future, with more specific training, this technology could not just improve readability for humans, but also for HTR models and perhaps even reduce storage or bandwidth requirements. Stay tuned for future updates and other insights into our

technology development on readcoop.eu/insights (<https://readcoop.eu/insights>).



Account ▼



(<https://readcoop.eu>)

Share this article:



Get started with Transkribus

Make your historical documents accessible



Learn more

(<https://readcoop.eu/transkribus/>)

The COOP

About us (<https://readcoop.eu/about/>)

Join us! (<https://readcoop.eu/join/>)

Our Members (<https://readcoop.eu/members/>)

Success Stories (<https://readcoop.eu/success-stories/>)

Work with us (<https://readcoop.eu/work-with-us/>)

Products & Services

Transkribus (<https://readcoop.eu/transkribus/>)

Transkribus.me (https://transkribus.eu/lite)
 Read&search (https://readcoop.eu/readsearch/)
 (https://readcoop.eu)
 ScanTent (https://readcoop.eu/scantent/)



Account ▼



Useful information

News (https://readcoop.eu/news/)

Download Transkribus (https://readcoop.eu/transkribus/download/)

Public Models (https://readcoop.eu/transkribus/public-models/)

Payment and shipping (https://readcoop.eu/payment-and-shipping/)

Helpful resources

Resource center (https://readcoop.eu/transkribus/resources/)

How-to Guides (https://readcoop.eu/transkribus/resources/how-to-guides/)

Getting started with Transkribus (https://readcoop.eu/transkribus/start/)

FAQs (https://readcoop.eu/transkribus/questions/)

Videos (https://readcoop.eu/transkribus/resources/videos/)





Account ▼



(<https://readcoop.eu>)

Copyright © 2023 READ-COOP SCE

[Terms & conditions \(https://readcoop.eu/terms-and-conditions/\)](https://readcoop.eu/terms-and-conditions/)

[Privacy Policy \(https://readcoop.eu/privacy-policy/\)](https://readcoop.eu/privacy-policy/)

[Contact \(https://readcoop.eu/contact/\)](https://readcoop.eu/contact/)

[Imprint \(https://readcoop.eu/imprint/\)](https://readcoop.eu/imprint/)

EN