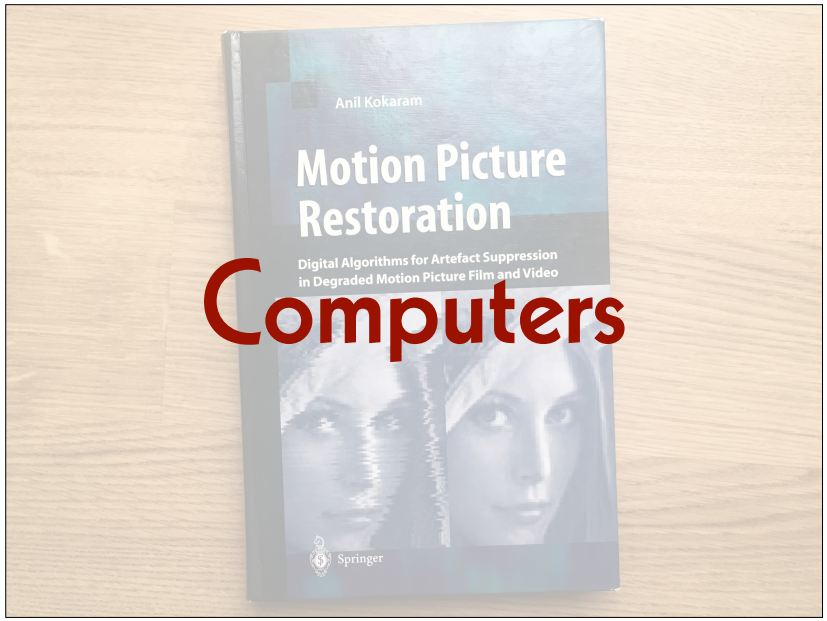
Deep machine learning applied to moving image restoration

Reto Kromer • AV Preservation by reto.ch

No Time to Wait 8
Zentrum für Kunst und Medien, Karlsruhe, Germany
11–13 September 2024

Summary

- computers
- artificial intelligence (AI)
- training
- lessons learned



In the beginning was fun …

1975
- programming on mainframe

1979
- building and programming own computers

1983
- first explorations of machine learning at EPFL

… then I fell under the spell …

1998
- Anil Kokaram: Motion Picture Restoration
- Kris Kolodziejski (1957–2012) and his Digital Film Lab, København

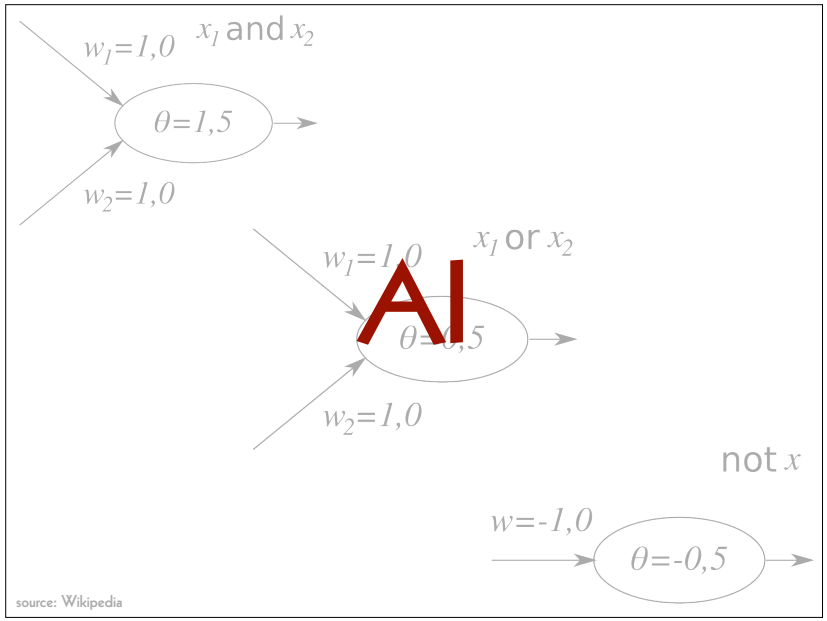
2001
- "Diamant" film restoration software
- "Preserve Then Show" conference in København

… and finally came profit

2004
- AV Preservation by reto.ch

2014
- late summer school "Beyond Black and White: Additive Colour Systems"

2017
- machine learning applied to restoration

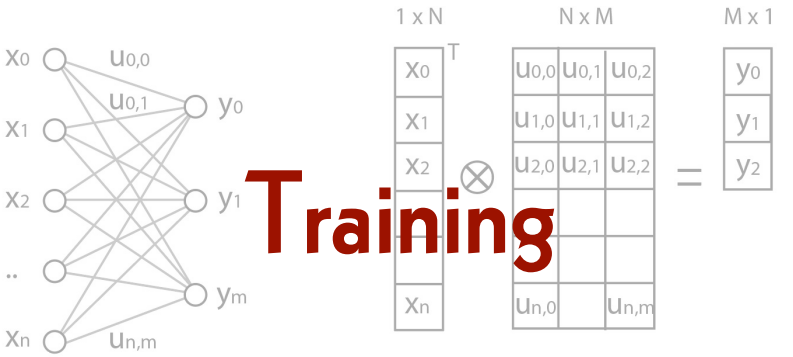


Definitions

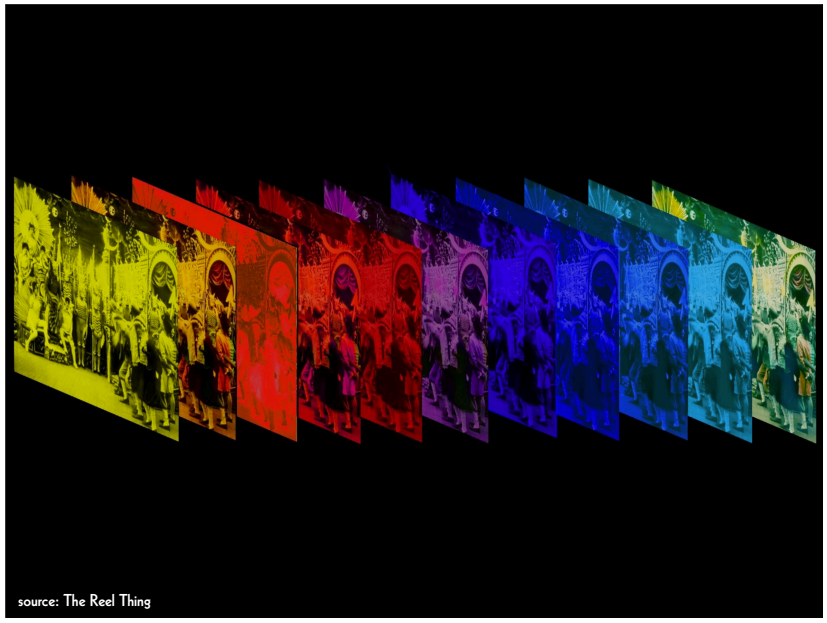
➔ please follow other presentations ;-)

Requirements

- ability to learn
- ability to deal with uncertainties and probabilities
- ability for humans to intervene and correct machine errors



source: Wikipedia



source: The Reel Thing

Deep machine learning

- organise data in multidimensional arrays
- operations can be expressed in terms of matrix multiplication and Kronecker product
- require a lot of GPU computing power

Data-based decision making

- detected anomalies are fixed via reinforcement learning

13

Build OpenFX libs and examples passing

OpenFX image processing plug-in standard

The authoritative source for information about OFX is http://openeffects.org/

- OpenFX Build Instructions
- OpenFX Documentation - start here
- OpenFX Documentation: Reference
- Programming Guide By Example
- OpenFX Wiki

Here are some Ways to get involved with OpenFX.

Why a Standard?

VFX plug-in vendors were frustrated for years because host application vendors created proprietary plug-in interfaces. As a result, each plug-in vendor had to port their plug-ins to all the different hosts and hosts couldn't use each other's plug-ins, limiting the selection of effects available to artists. The need for a standard interface was clear, so Bruno Nicoletti of The Foundry led the effort to develop a standard. That standard is OFX.

OFX is a win for artists because there is no waiting for plug-in vendors to port their cool effects to your application. Once a host compositing or editing application adopts OFX, all OFX plug-ins on the market instantly become available on that host.

source: github.com/AcademySoftwareFoundation/openfx

14

Building OpenFX: Libs and Plugins

OpenFX itself is only a set of C header files, the ones in include. This repo also includes the C++ support lib, giving a C++ API on top of the basic C, and two sets of example plugins; one set with the support lib, one set using the raw C API. There is also a host support lib for use when creating a new OpenFX host. These instructions show how to build the support libs and all the plugins, and install them into your plugin folder.

Prerequisites

OpenFX uses cmake and conan to build. Other dependencies are fetched by conan. The build requires Conan 2.1.0 or later, and CMake 3.28 or later.

Install cmake:

- Mac: `brew install cmake`
- Windows: `choco install cmake`
- Linux: `apt install cmake`

Install conan (version >= 2.1.0 recommended) using pip (and python3)

- `python3 -mpip install 'conan>=2.1.0'`

Standard Builds

To build and install everything use scripts/build-cmake.sh

source: github.com/AcademySoftwareFoundation/openfx

15

Lessons learned

source: Wikipedia

16

Pros

- improves computer performance
- opens new human-machine interactions
- processes information faster than humans

Cons

- implementation costs are high
- software development is expensive and the necessary development resources are scarce
- skilled programmers almost inexistent

Conclusions

- Without using this technology, we would never have been able to realise certain of our projects.
- It was a lot of work …
- … and there was a lot of fun!

AV Preservation by
reto.ch

Sandrainstrasse 3
3007 Bern
Switzerland

reto.ch
info@reto.ch