

# Deep machine learning applied to moving image restoration

Reto Kromer • AV Preservation by reto.ch

## **On the Materiality of Audio-Visual Heritage**

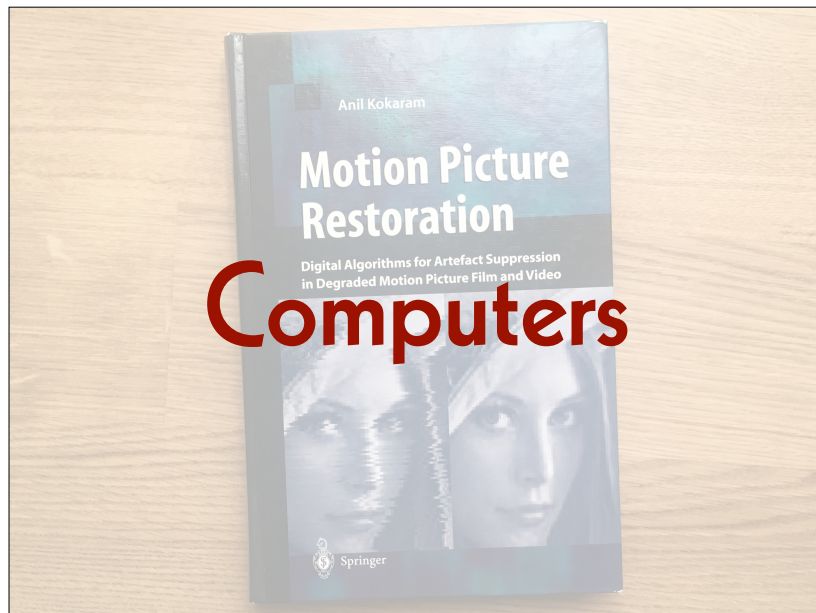
Elías Querejeta Zine Eskola  
Donostia (San Sebastián), Spain  
15–18 October 2024

1

## Summary

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

2



3

## In the beginning was fun ...

1975

- programming on mainframe

1979

- building and programming own computers

1983

- first explorations of machine learning at EPFL

4

## ... 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

5

## ... 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

6

### ๒. สิทธิของอนุชน

- ๒.๑ ด้วยตระหนักในความรับผิดชอบของตนในอนาคตที่จะอนุรักษ์ฟิล์มภาพยนตร์ให้ดำรงอยู่อย่างยั่งยืน หอภาพยนตร์จักยืนหยัดต่อการป้องกันการบีบบังคับใด ๆ ที่จะกำจัดหรือทำลายสิ่งของที่หอภาพยนตร์ซึ่งมีคุณค่าอันเลือกที่จะปฏิเสธหรือรับสงของใด ๆ ที่มีผู้เสนอให้ขอเก็บสะสมด้วยเหตุผลใดซึ่งอาจอยู่นอกเหนือกฎเกณฑ์ว่าด้วยการอนุรักษ์หรือนโยบายการคัดเลือกที่ใช้อยู่ของหอภาพยนตร์นั้น

### ๓. สิทธิในการใช้ประโยชน์

- ๓.๑ หอภาพยนตร์ตระหนักว่า สิ่งของในความดูแล มีทั้งมูลค่าในเชิงพาณิชย์และคุณค่าในทางศิลปะ สิ่งนี้ให้อรรถประโยชน์แก่ผู้ชม

# Principles

7

## Conservation

Conservation encompasses all activities for the care of an object, which **delay its further decay** and ensure that it remains in the most intact condition for the future.

8

## Restoration

Restoration includes all interventions and treatments that serve to **retrieve a certain historical state** and contribute to the legibility, aesthetic integrity or **reuse** of an object.

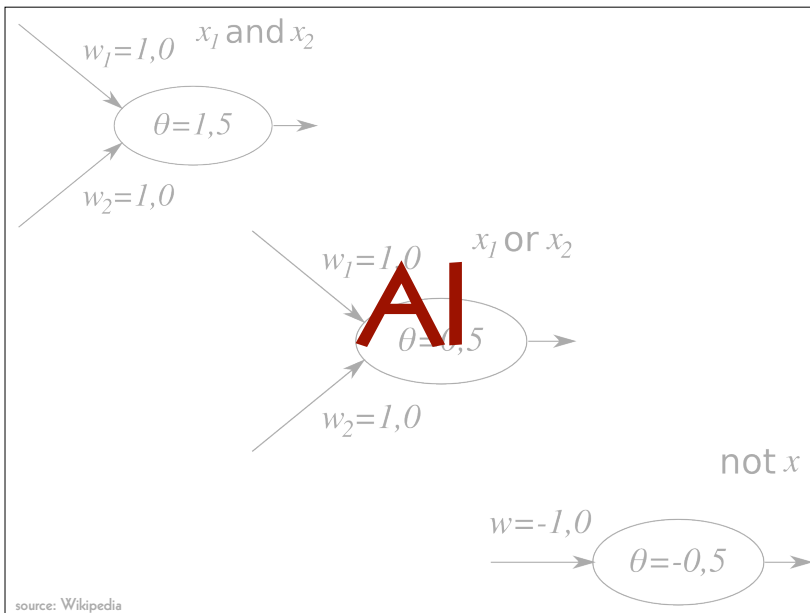
Restorative actions may be irreversible and require great care in planning, justification, execution and **documentation**.

9

## An ounce of ethics

- The probability that a work is available in its integrity in the future is increased.
- All the options that existed before taking an action remain open after the action.
- Every step is carefully documented.

10



11

## Definitions

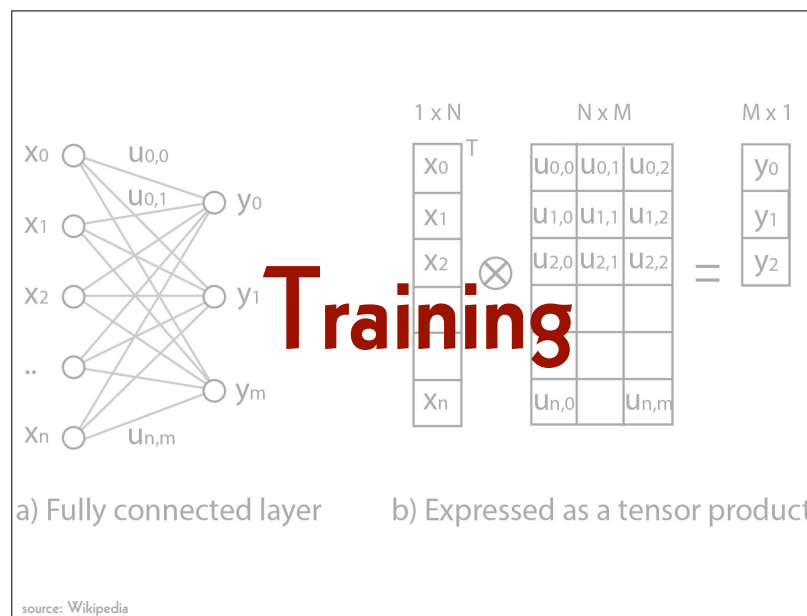
- ...
- ...

12

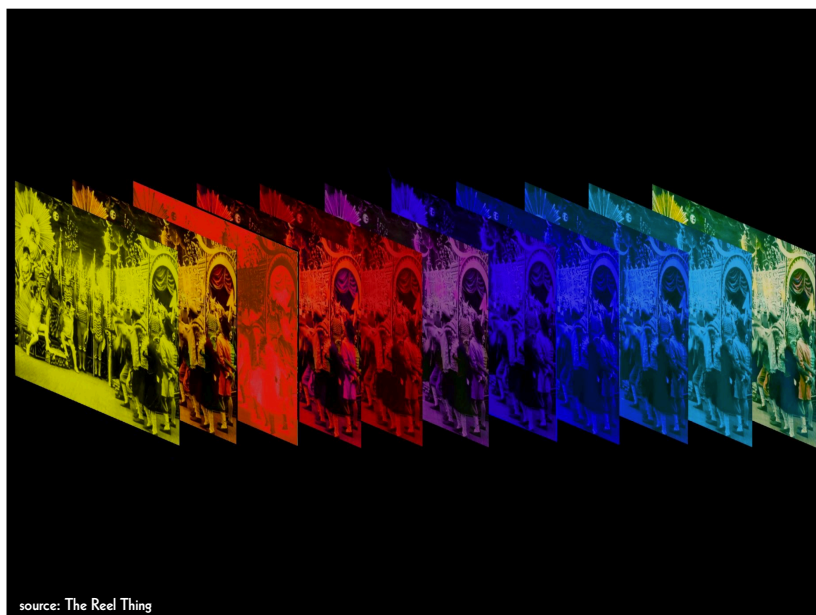
## Requirements

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

13



14



15

## 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

16



# Data-based decision making

- detected anomalies are fixed via reinforcement learning

17

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](https://github.com/AcademySoftwareFoundation/openfx)

18

## 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](https://github.com/AcademySoftwareFoundation/openfx)

19



source: Wikipedia

20

## Pros

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

21

## Cons

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

22

## 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!

23

AV Preservation by  
**reto.ch**

Sandrainstrasse 3  
3007 Bern  
Switzerland

reto.ch  
info@reto.ch



24