Optical Music Recognition

Jorge Calvo Zaragoza

Sound and Music Computing Summer School
Málaga (Spain) May 26, 2019

Speaker

Jorge Calvo Zaragoza (jcalvo@dlsi.ua.es)

Assistant Professor at University of Alicante, Spain

Background

- *PhD in Computer Science* from University of Alicante, Spain
- *Postdoctoral Fellowship* at McGill University, Canada
- *Postdoctoral Fellowship* at Universitat Politècnica de València, Spain

Speaker

Jorge Calvo Zaragoza (jcalvo@dlsi.ua.es)

Assistant Professor at University of Alicante

Research topics:

- Machine Learning
- Deep Learning
- Computer Vision
- Document Image Analysis
- Optical Music Recognition

No music background!

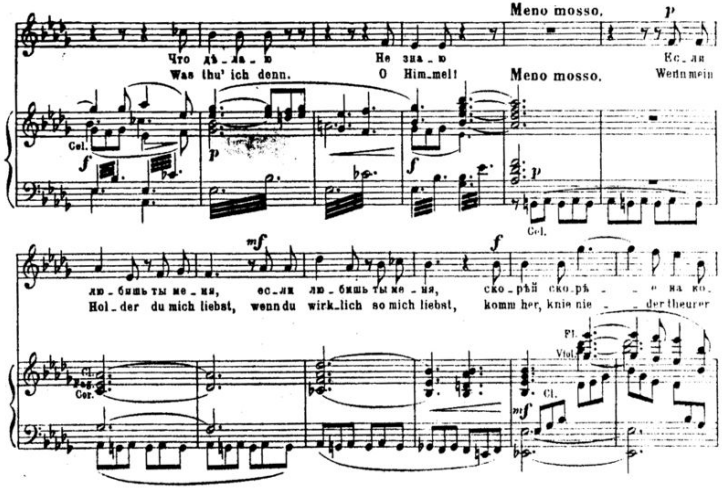
Outline

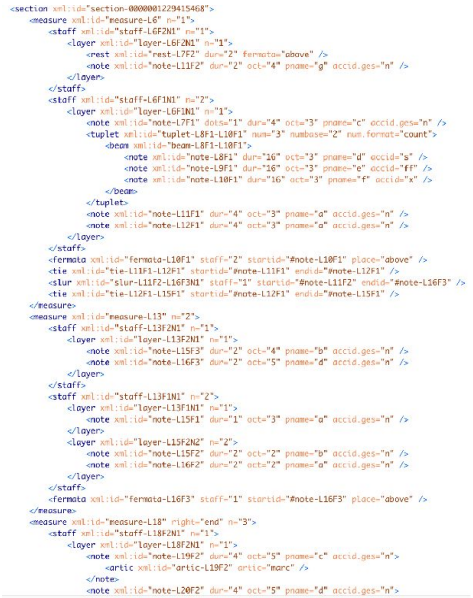
- **Session I:** Introduction and definitions
- **Session II**: Approaches and state of the art
- **Session III**: Step-by-step OMR system with Keras
- **Session IV:** Closing remarks

Session I
Introduction

Introduction

What Optical Music Recognition (OMR) does?





Introduction

What is OMR for?

Introduction

The field is appealing from different points of view.

- Potential: it enables any computational process over written music.
- Scientific: it involves pattern recognition, image processing, digital libraries, natural language processing, and musicology.

Motivation

- Most written music has never been recorded or stored in a structured format that allows further computational processes.
- Typesetting sheet music is a extremely time-consuming task.
- OMR represents a key element to diversifying the sources over which computational music processes operate.

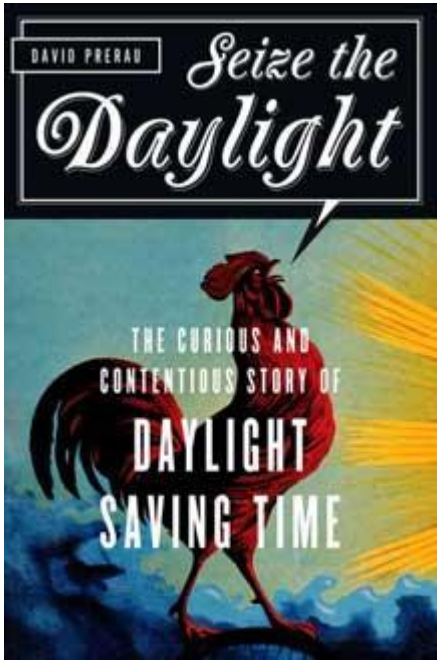
History of OMR

History

First attempts

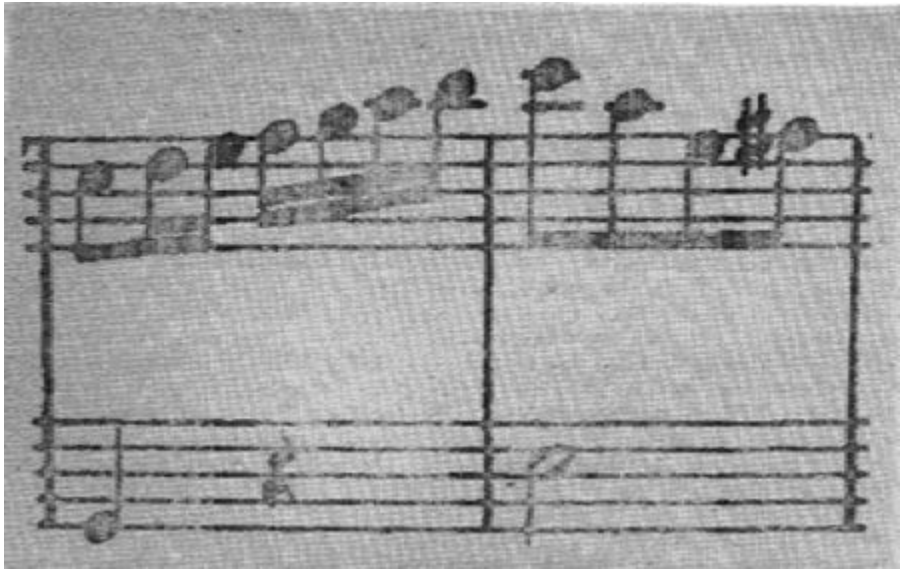
- Dennis Pruslin: “Automatic recognition of sheet music”
  - Sc.D. Dissertation, MIT 1966
- David Prerau: “Computer pattern recognition of standard engraved music notation”
  - Ph.D. Dissertation, MIT 1970

Why MIT?



History

The first scanned music score that was published! (Prerau, 1970)

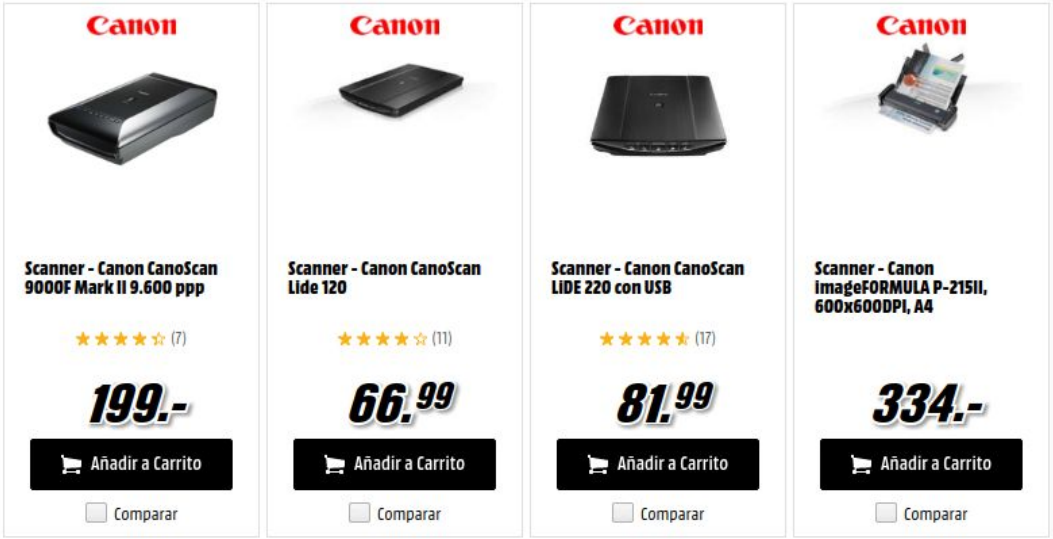


History

Scanner at the United States Census Bureau (~1960)

History

Currently…



History

Dissertations about OMR:

- 1966 Denis Pruslin (MIT: *MSc*)
- 1970 David Prerau (MIT)
- 1989 Nicholas Carter (University of Surrey)
- 1996 Ichiro Fujinaga (McGill University)
- 1996 Kia Ng (University of Leeds)
- 1996 Bertrand Coüasnon (Université de Rennes)
- 1997 David Bainbridge (University of Edinburgh)
- 2006 Laurent Pugin (Université de Genève)
- 2009 Alicia Fornés (Universitat Autònoma de Barcelona)
- 2012 Ana Rebelo (Universidade do Porto)
- 2014 Andrew Hankinson (McGill University)
- 2016 Jorge Calvo Zaragoza (Universitat d'Alacant)

Under development:

- Jan Hajič Jr. (Charles University)
- Alexander Pacha (TU Wien)
- Kwon-Young Choi (Université de Rennes)
- Arnau Baró (Universitat Autònoma de Barcelona)
- Francisco J. Castellanos (Universitat d'Alacant)

More than 1,000 dissertations about text recognition from images

What is Optical Music Recognition?

What is Optical Music Recognition?

- People share a common understanding of what OMR is, vaguely defined as “reading music notation with a computer”.
- However, there is a lack of elaboration of the actual meaning of the discipline
- Most scientific research focuses on the solution to certain (sub-)problems, and so the meaning of the discipline is not that relevant.

What is Optical Music Recognition?

- A critical review of the literature reveals a great variety of definitions that can be broadly grouped into two extremes:
  - Definitions to motive the work at hand:
    - Example: “Convert sheet music images to MIDI files”
  - Rather generic:
    - Example: “OCR for music”

What is Optical Music Recognition?

- People rely on their intuition to compensate for this lack of accuracy.
- A definition that describes the whole OMR essence is:

*"**Optical Music Recognition** is the **field of research** that studies how to **computationally read music** notation in **documents**"*

What is Optical Music Recognition?

- The first part of the definition makes a strong emphasis on OMR as a *research field,* instead of “task” or “process”
  - OMR cannot be properly formalized in terms of unique inputs and outputs.
  - Within this research field, several tasks can be formulated with specific, unambiguous input/output pairs.

What is Optical Music Recognition?

- The term *computationally* distinguishes OMR from the musicological and paleographic studies of how to decode a particular notation system.
- It also excludes studying how humans read music:
  - OMR does not study the music notation systems themselves;
  - Instead, it builds upon this knowledge, with the goal that a computer should be able to read the music notation as well.

What is Optical Music Recognition?

- The last part of the definition “*reading music notation in documents”* tries to define OMR in a concise, clear, specific, and inclusive way.
- To fully understand this part of the definition, it is necessary to clarify what kind of information is captured in a music notation document and outline the process by which it gets generated.
- Then we can elaborate on how OMR attempts to **invert this process** to read and recover the encoded information.

What is Optical Music Recognition?

From *music* to *music score*

- Music can be conceptualized as a **set of notes in time.**
  - A note is a musical object that is defined by four parameters:
    - *pitch*, *duration*, *loudness*, and *timbre.*
  - Additionally, it has an *onset*: a placement onto the time axis.
    - In music it does not mean wall-clock time, but measured in beats.
- Notes are grouped hierarchically into phrases, voices, and other musical units that have logical relationships to one another.
- This structure is a vital part of music: it is essential to work it out for making a composition comprehensible.

What is Optical Music Recognition?

From *music* to *music score*

- To represent this conceptualization of music visually, so that it can be interpreted as the composer conceives it, many notation systems have been developed and evolved over time.
- Music notation is a **visual language** that **encodes music** in a **graphic form**, and complements it with information on how to interpret and play it (articulations, dynamics, etc.).
- Like any other language, it comprises a series of symbols (alphabet) and rules on how to position these elements (syntax) to capture the concept to be transmitted (semantics).

What is Optical Music Recognition?

From *music* to *music score*

- Unfortunately, all music notation systems entail a certain loss of information: they are designed to preserve the most relevant properties of the composition but other aspects are omitted.
- It is the interpreter's responsibility to fill in those omissions appropriately.

What is Optical Music Recognition?

From *music* to *music score*

- Unquestionably, the predominant music notation at the moment is the western modern notation (Common Western Music Notation, CWMN).
- However, there is still a significant amount of music manuscripts written in other notations.

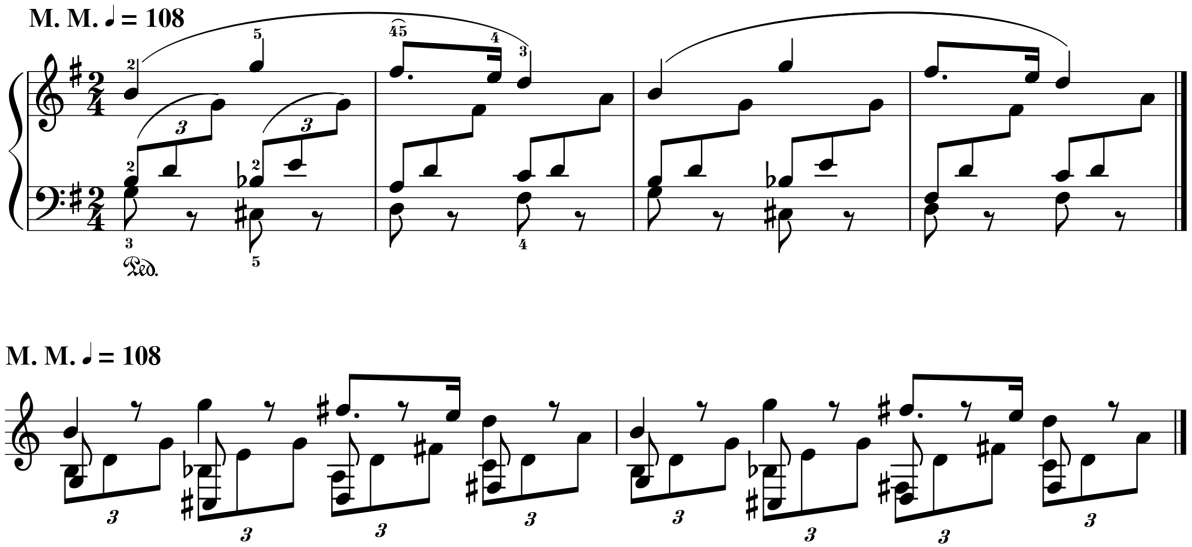
What is Optical Music Recognition?

From *music* to *music score*

- Once a notation is chosen to write the piece, there is still the challenge of choosing the most appropriate *engraving*: a set of notes can be expressed in many different ways.
  - One criterion could be to make the notation as "readable" as possible.
  - These decisions not only affect the visual appearance but also help to preserve the logical structure.

What is Optical Music Recognition?

From *music* to *music score*

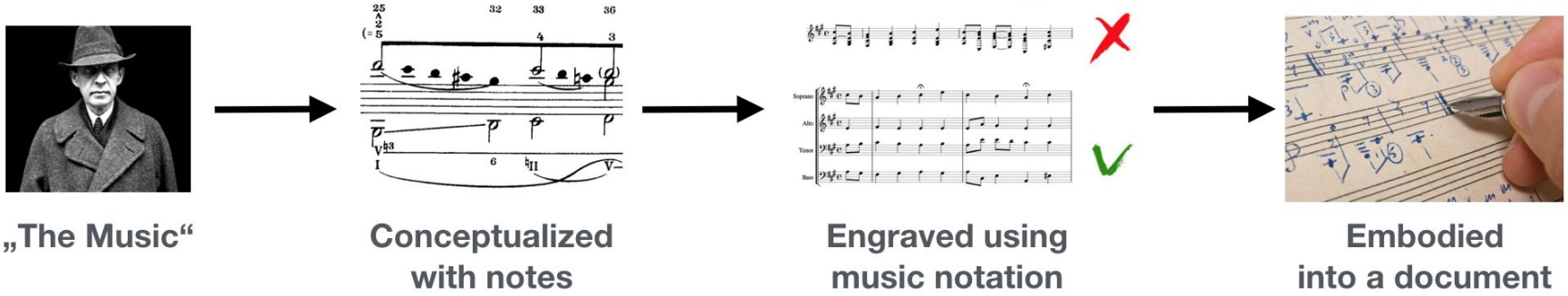


Robert Schumann: Kinderszenen, Op. 15, No. 1 "Von fremden Ländern und Menschen"

What is Optical Music Recognition?

From *music* to *music score*

- Finally, the music notation is embodied in a document:



- OMR can be understood in terms of **inverting** this process with a computer.

What is Optical Music Recognition?

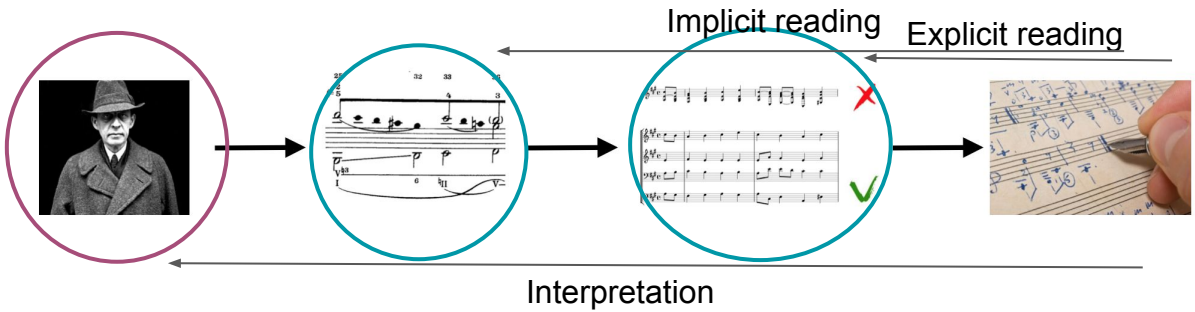
Inverting the music encoding process

- Following our elaboration, there exist two ways of interpreting the term “reading” music scores:
  - **Recover music notation** and information from the engraving process, i.e., what elements were selected to express the given piece of music and how were they laid out?
    - This does not necessarily require specific musical knowledge, but it does require an output representation that is capable of storing music notation, e.g., MusicXML.
  - **Recover musical semantics**, which we defined as the notes, represented by their characteristics.
    - In practical terms, MIDI would be an appropriate output representation for this goal.

What is Optical Music Recognition?

Inverting the music encoding process

- This is a fundamental distinction that dictates further system choices
  - Explicit reading: recover music notation
  - Implicit reading: recover musical semantics



What is Optical Music Recognition?

Inverting the music encoding process

To sum up, OMR can:

- Process the document in terms of music-notation symbols.
- Use the music notation to decode the notes and their characteristics.

But OMR does not *interpret*:

- It cannot fill the gaps that the music notation does not capture.
- In other words, it ends where performers may start to disagree.

What is Optical Music Recognition?

Inverting the music encoding process

- In addition, there are other reasons for reading music scores:
  - Paleographical analysis
  - Retrieve metadata
  - Content-based search in written sources

- We therefore need to broaden the scope of OMR to actually capture these applications as well, according to the proposed definition.

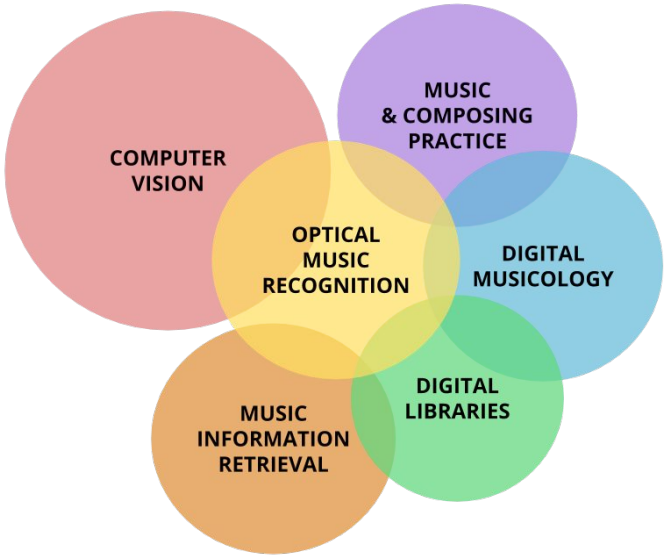
What is Optical Music Recognition?

Relation with other disciplines

- OMR can be put in context of other disciplines
  - From a technical point of view, OMR can be considered a subfield of **computer vision and document analysis**.
  - In the context of **digital libraries and computational musicology**, OMR allows access to compositions for which only written scores are available (probably most of the musical heritage).
  - OMR has practical implications for **composers and performers**, since it reduces the costs of digitizing scores and bringing the benefits of digital formats to daily practice.

What is Optical Music Recognition?

Relation with other disciplines



What is Optical Music Recognition?

Relation with other disciplines

- Some other questions must also be addressed:
  - From a computer science point of view, why should OMR be studied in addition to optical character recognition (OCR)?
  - What is the justification for speaking specifically about music notation, in the context of other graphic recognition domains?
  - Are there special considerations in OMR that one does not find in other writing systems?

What is Optical Music Recognition?

Relation with other disciplines

- A part of the justification lies in the properties of musical notation as a writing system.
- Unlike most natural-language writing systems, the alphabet of musical notation is **structural**.
  - Its alphabet consists of well-defined primitives but their structure and configuration (how they are placed and organized in the staves) is what determines their meaning.
  - The note-heads, stems or beams do not have meaning isolatedly.

What is Optical Music Recognition?

Relation with other disciplines

- This implies that musical notation has to be processed beyond its basic primitives, since otherwise it has no meaning.
- There is not a good equivalent of this step of interpretation in OCR, since it is not necessary to retrieve how a configuration of symbols (characters) should be interpreted.
    - This is left to humans or other well-defined tasks in the field of natural language processing.

WIKIPEDIA

Die freie Enzyklopädie

Deutsch
2 218 000+ Artikel

English
5 714 000+ articles

Español
1 472 000+ artículos

日本語
1 119 000+ 記事

Русский
1 495 000+ статей

Français
2 039 000+ articles

Italiano
1 459 000+ voci

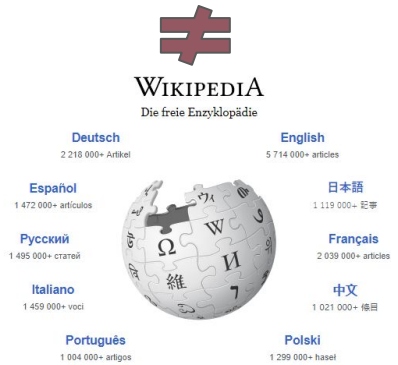
中文
1 021 000+ 條目

Português
1 004 000+ artigos

Polski
1 299 000+ haseł

Wikipedia
Die freie Enzyklopädie

≠



What is Optical Music Recognition?

Relation with other disciplines

- Since music can be conceptualized as notes, and notes are well-defined objects that can be retrieved from the score, an OMR system is required to produce this additional level of expected (and achievable) results that OCR does not generate.
- In a way, OMR is more ambitious than OCR, since there is an additional interpretation step, specifically for music, that does not have a good analogy in other domains.

What is Optical Music Recognition?

Relation with other disciplines

- Other distinctive features of music notation are:
  - Complex graphic alphabet
  - Strong imbalance in the occurrence of primitives
  - Symbols that vary drastically in shape and size
  - Two-dimensional syntax
    - Left to right
    - Top to bottom

Uses of OMR

Uses of OMR

- Now that the meaning of "a computer that reads music scores" has been addressed, we can reexamine what it should be attempted for.
- We all know **why**: the computer automatizes the task: faster and cheaper

- But, what are the applications of a computer that “reads scores”?
- Generally, we can organize the applications by the **type of information** that we get as a **result**.

Uses of OMR

Uses of OMR, grouped by requirement:

- **Structured encoding**: obtain a structured representation of musical notation.
  - *How was the score written in terms of music notation?*
- **Replayability**: get a structured representation of the written music.
  - *What music is being represented?*
- **Search**: search and retrieval of content.
  - *Given a (music) query, where is it?*
- **Document metadata extraction**: analysis of the musical document as an entity.
  - *Questions about the score itself, from a (paleo)graphic point of view.*

Uses of OMR


Document Metadata Extraction
Search
Replayability
Structured Encoding
Partial
Complete
Level of Comprehension

Uses of OMR

- Another way of understanding the previous taxonomy is by the evaluation of the system that is built for each requirement.
- (Automatic) evaluation is a key aspect in computer science, since it allows large-scale experiments that evaluate the performance of different proposals to solve the same problem.
- We will see that all the applications of each specific use are evaluated in a similar way.

Uses of OMR

Document metadata extraction

Uses of OMR: Document metadata extraction

- The first application area corresponds to one in which musical documents are processed graphically to **answer specific questions about themselves**.
- For example:
  - Is there music notation in the image?
  - Which period the piece was written in?
  - What kind of notation was used?
  - Who is the writer of the piece? (not the composer!)

Uses of OMR: Document metadata extraction

- These examples entail a different underlying complexity from a computational point of view.
- However, they can be formulated in a simple way, regardless of their complexity: **classification** tasks.
  - The answer is a category within a predefined set.
- In these cases, it is easy to measure the quality of the system's performance based on its success or error rate.

Uses of OMR

Search

Uses of OMR: Search

- We define **search** as the application in which the main question is **where a specific content is located**.
- All tasks can be formulated with the same result: the location of the information given as query.
- In this case, there are several levels of specification, since "location" is a loose concept:
  - Piece
  - Page
  - Staff
  - Bounding box (pixels) of an image

Uses of OMR: Search

- Information retrieval scenarios, such as finding handwritten copies or tracking musical themes in score collections, have less complexity than recognizing content (music or notation).
- It is not necessary to understand the musical notation itself for a search application to reasonably find a content.
- In this context, the recognition system works basically as a descriptor: one can discard complex elements of the score and still obtain useful descriptors that allow finding a content.

Uses of OMR: Search

- Therefore, the advantage of this application is that it does not require absolute precision to make its operation interesting and useful.
- Several types of search can be proposed:
  - Search by content (symbolic)
  - Search by graphic extract (copies)
  - Search by audio (multi-modal retrieval)
- The result is not always unique, but the system may return all content that matches the search criteria.

Uses of OMR: Search

- The system is not explicitly evaluated in its ability to extract music or describe the musical notation elements used.
- These tasks are evaluated based on two concepts:
  - Precision: the elements found meet the criteria.
  - Recall: the elements that meet the criteria are found.
- These applications usually work with thresholds
  - What confidence does the system need to effectively recover the content?
  - Precision and recall are mutually exclusive with respect to the threshold value.

Uses of OMR

Replayability

Uses of OMR: Replayability

- The replayability application is related to the reconstruction of the notes encoded in the score (implicit reading).
- It can be defined as the necessary recognition to **create an audible version** out of the score given to the system.
- Producing audio is not the specific goal, although it is certainly an attractive application in many contexts.
- This symbolic representation (generally understood as a MIDI file) is already a very useful abstraction of the score itself.

Uses of OMR: Replayability

- Obtaining a "reproducible" file enables the use of computational tools (audio synthesis, musical similarity functions, melodic analysis, etc.) that are possible on music but not on written sources.
- Producing a MIDI representation (or equivalent) is, at least in the foreseeable future, a key objective for OMR.
    - MIDI is a representation of music that already has a long tradition in computer processing of music, with a wide variety of purposes.

Uses of OMR: Replayability

- Given the cost of manual typesetting, an OMR that produces MIDI is probably the only tool that makes the large number of compositions available for quantitative musicological research.
  - Especially if it can do so for handwritten notation.
- One could begin to answer broad questions about the evolution of musical styles, instead of simply relying on the works of the relatively few known composers.

Uses of OMR: Replayability

- From a scientific point of view, an advantage of this application is that it is evaluable
  - There are well-known algorithms to compute the difference between the reproduction retrieved by the system and the ground-truth one.

Uses of OMR

Structured encoding

Uses: structured encoding

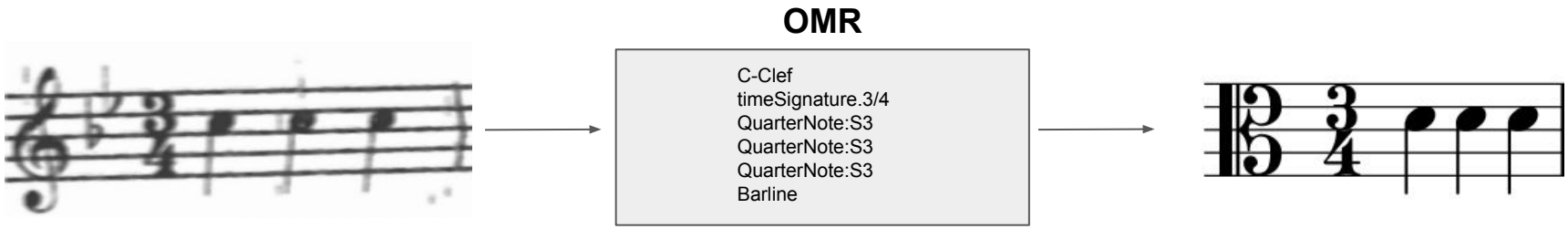
- Structured encoding applications aim to preserve all available information of the musical score (explicit reading).
- Essentially: transcribing the document into a structured digital format, with the ultimate goal of **maintaining the same information** that could be recovered from the physical source.
    - For instance, converting the input to MEI or MusicXML.

Uses: structured encoding

- Since music editing tools are tedious and time-consuming, automatic technology represents the only alternative for obtaining the structured coding of a large number of scores at a reasonable time and cost.
- This scenario has been the **main motivation** in the majority of works related to OMR.
- From a scientific point of view, it is the task that represents the greatest challenge since it is at the highest end of complexity.

Uses: structured encoding

- It is not clear how to evaluate OMR for this application



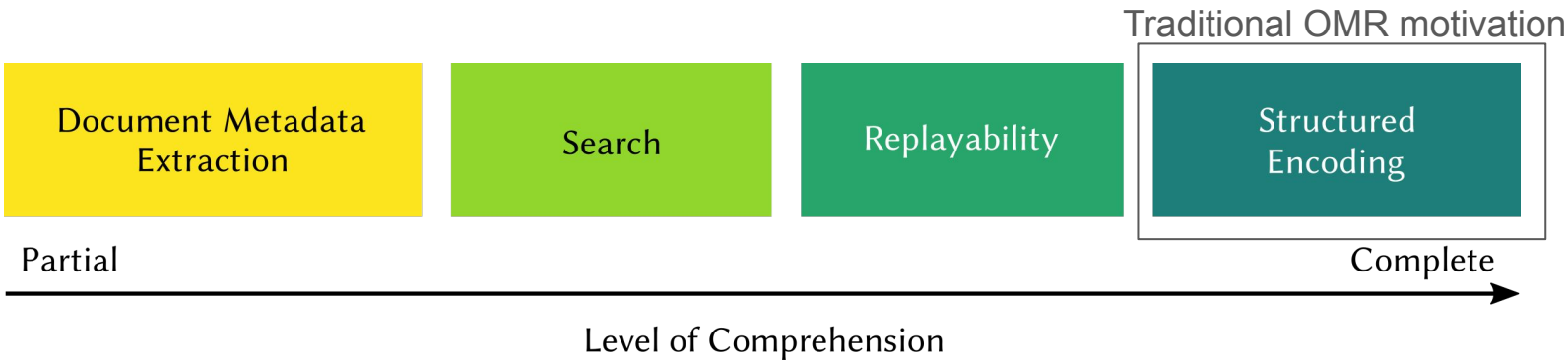
Hoy many errors would you count?

- This is one of the **open questions** of the OMR field

“Credit for errors” between graphics and semantics

- Errors at different levels interact
  - Undetected 8th flag implies wrong duration
- The interaction is not straightforward
  - False positive beam instead of slur influences multiple notes
- The interaction has scope partly independent on location of error
  - Wrong clef affects many pitches
  - False positive barline cancels inline accidentals early

Uses: structured encoding



The OMR domain

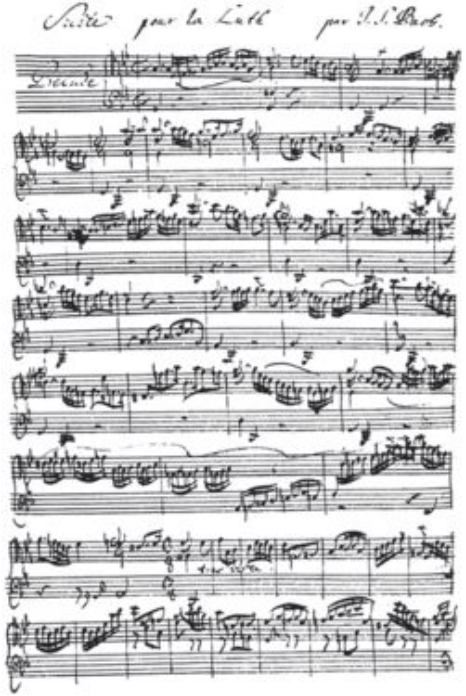
Input domains

"OMR is an interesting idea but *it does not work*: I tried *PhotoScore* and it completely failed with my *handwritten* scores from the *XVIIIth century*"

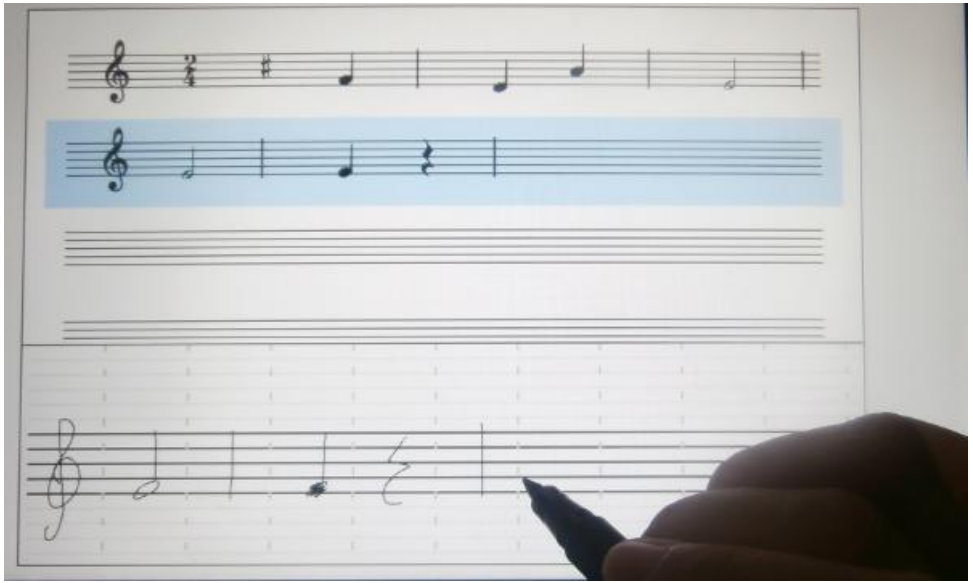
Anonymous librarian

At the Music Encoding Conference, Washington, 2018

Input domains: type of the signal

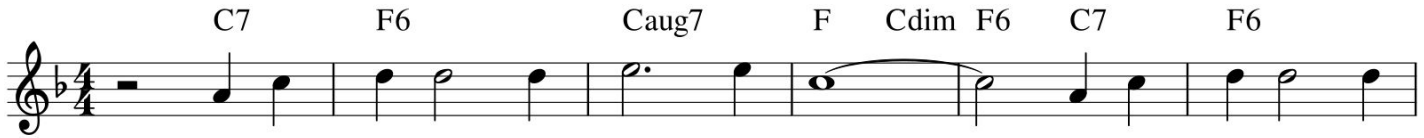


Offline: score image

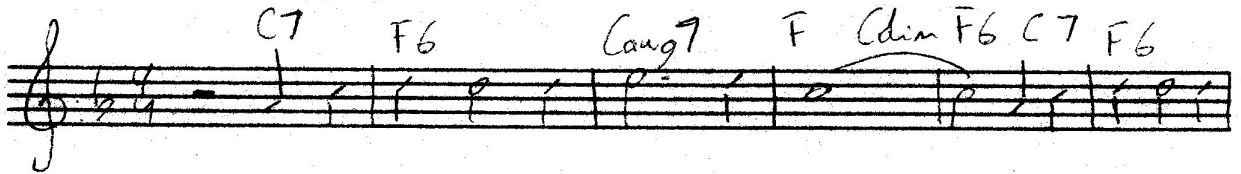


Online: pen strokes

Input domain: engraving mechanism

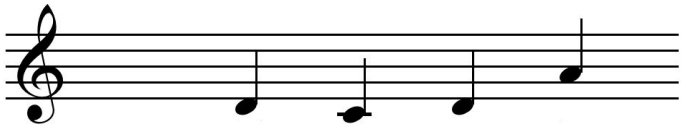


Typeset

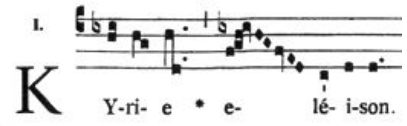


Handwritten (typically over uniform staff lines)

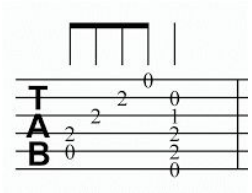
Input domain: notation



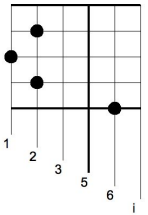
Common Western Music Notation



Preceding notations (mensural, neumes, ...)



Instrument-specific notation (tablature, drums, ...)

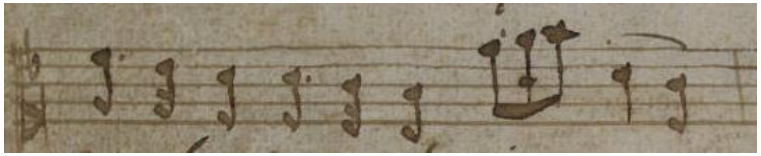


Others (Numbered, Yogyakarta, Surakarta, ...)

Input domain: graphical complexity

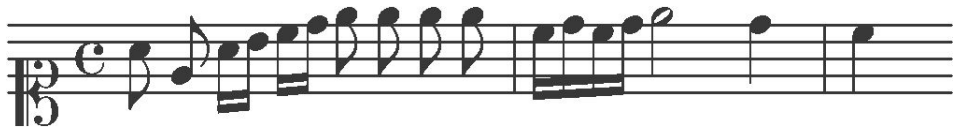
Ideal conditions

Camera-based issues

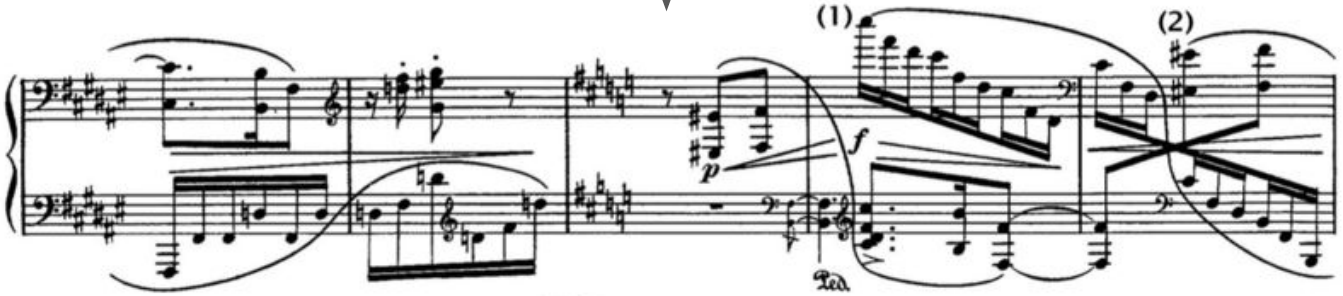


Degraded documents

Input domain: structural complexity



Single voice - Single staff



Multiple voices - Multiple staves

Closing

When you are not working on OMR...

"*Please, solve the Optical Music Recognition problem once and for all*"

Barbara Haws
Archivist and Historian of New York Philharmonic

Keynote lecture at the 17th International Society for Music Information Retrieval Conference, New York, 2016

When you work in OMR...

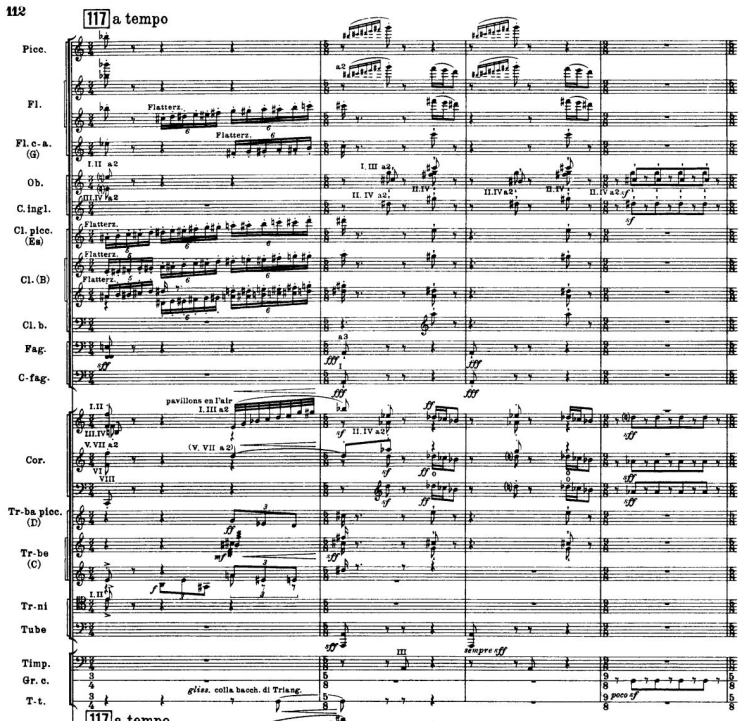
"Computers are pretty stupid and I'm not smart enough to teach them"

Ichiro Fuinaga
Chair of the Music Technology Area of the Schulich School of Music at McGill University

Interview for Calcul Quebéc, 2018

Why is OMR difficult?

- Music notation is extremely complex!
- It is a formal language but…
  - Quite often, its rules are ignored
    - Minor simplifications
      (e.g., 3's omitted over a triplet)
    - Big breaks (e.g., invalid metric)
    - Examples at [Donald Byrd's Gallery]



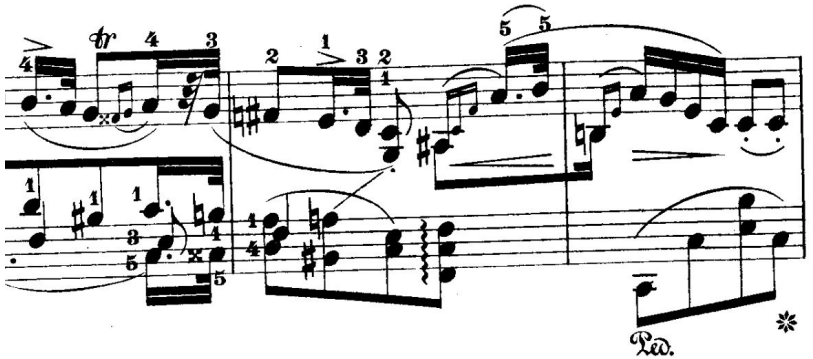
"The Rite of the Spring" de Igor Stravinsky

Why is OMR difficult?

Graphical complexity (e.g., the scores are dense, the symbols overlap, there are ambiguities)

Structural complexity

- Infinite possibilities
- Syntactic and semantic rules
- Rules can be broken!



“Nocturne (Op. 15, no. 2)” by Frédéric Chopin

➢ Glass ceiling: where the performers do not agree

Why is OMR difficult?

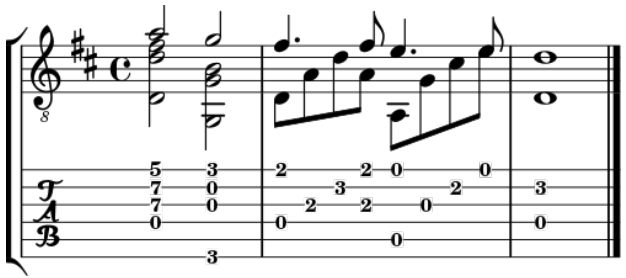
- Different types of notation with their own particularities
  - Example: in Mensural notation, the meaning of a glyph varies depending on its context, but also on the date the score was written.

Note values

Name		Century			
		13th	14th	15th	17th
Maxima	Mx				
Longa	L				
Breve	B				
Semibreve	Sb				
Minim	Mn				
Semiminim	Sm				
Fusa	F				
Semifusa	Sf				

From https://en.wikipedia.org/wiki/Mensural_notation

Guqin Notation from Wikipedia



From Wikipedia

Last remarks

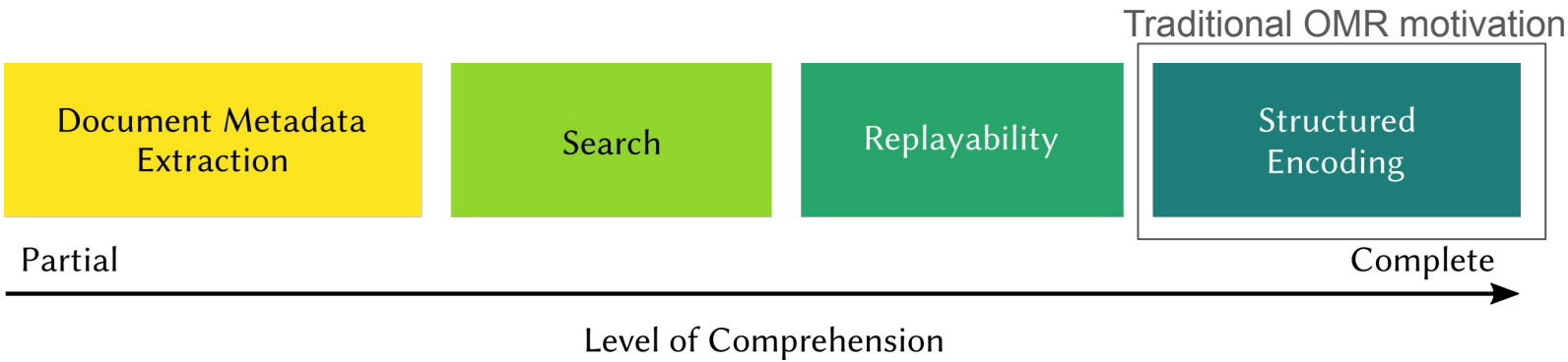
A **universal solution to OMR is out of reach** (with current technologies)

- As in many other fields of artificial intelligence, graphics recognition systems are designed for a particular input domain.
  - In the case of music scores, an OMR must be specific to a collection, or a set of collections with similar characteristics
  - At least, they must have the same notation symbols.
- The requirements vary depending on the application
  - OMR is not a single problem!
- More importantly, **OMR will never be perfect!**

Session II

Approaches and state of the art

OMR by applications

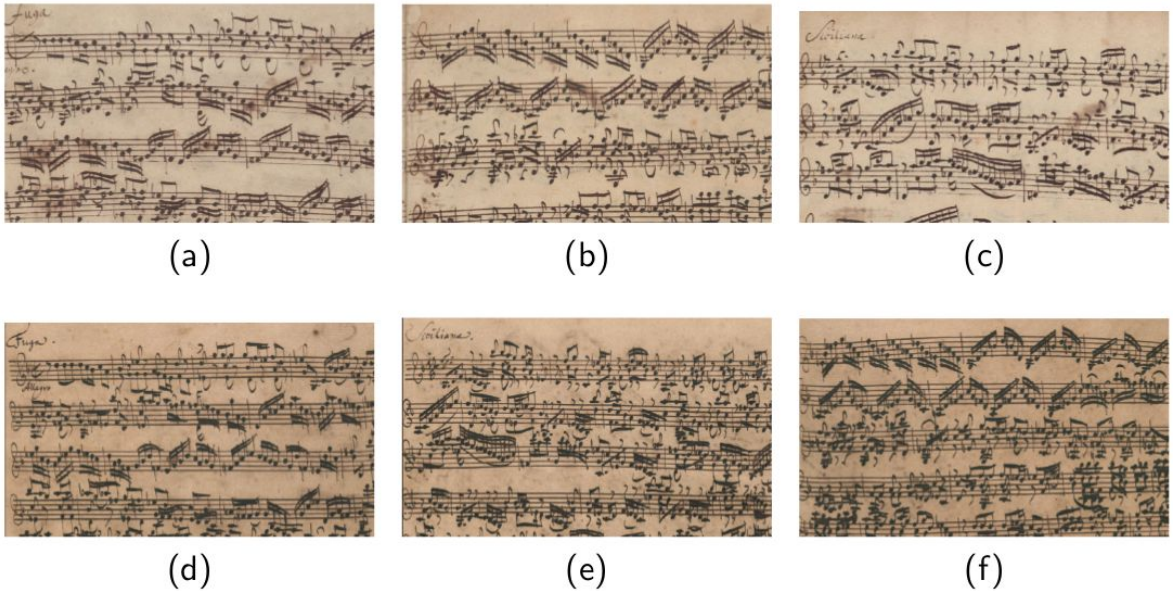


Approaches to OMR

- There are many specific methods for metadata and search applications, and their associated tasks, that will not be covered here.
- For example:
  - Writer identification
  - Pattern retrieval
    - By symbolic specification
    - By example
    - By audio

Approaches to OMR

Writer Identification: Johann Sebastian Bach y Anna Magdalena Bach



Approaches to OMR

- We will focus on **structured encoding** methodologies for scores given as **images**, as they require the highest level of comprehension and represent the most frequent input formulation.
- This research dates back to the 1960s
- There has been much independent research
  - Past research is covered in some surveys (Blostein and Baird, 1992; Rebelo et al., 2012)
- Few complete systems
  - Most contributions focused on solving (sub-)stages of the **OMR pipeline**

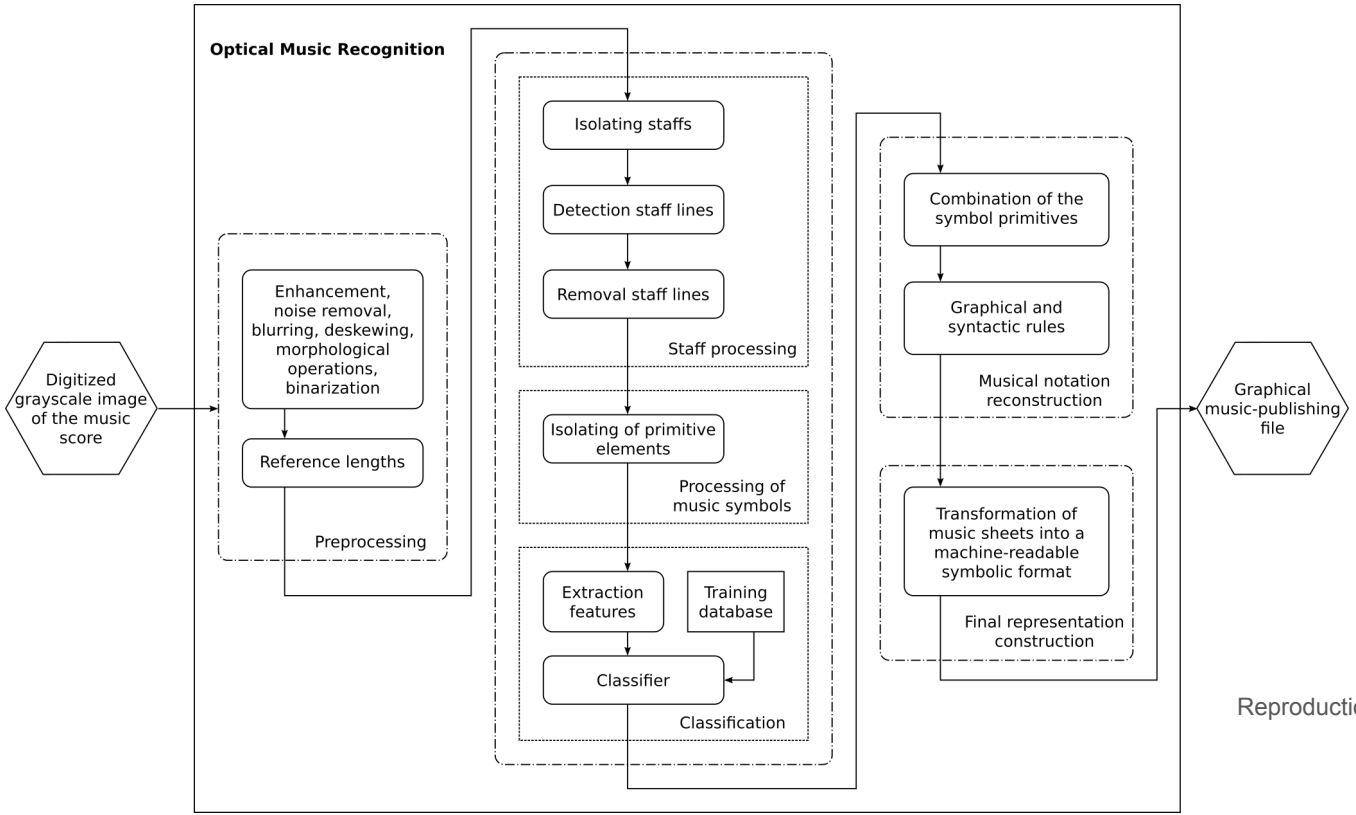
The OMR pipeline

The OMR pipeline

The traditional full-pipeline OMR approach consists of several independent stages that are performed sequentially:

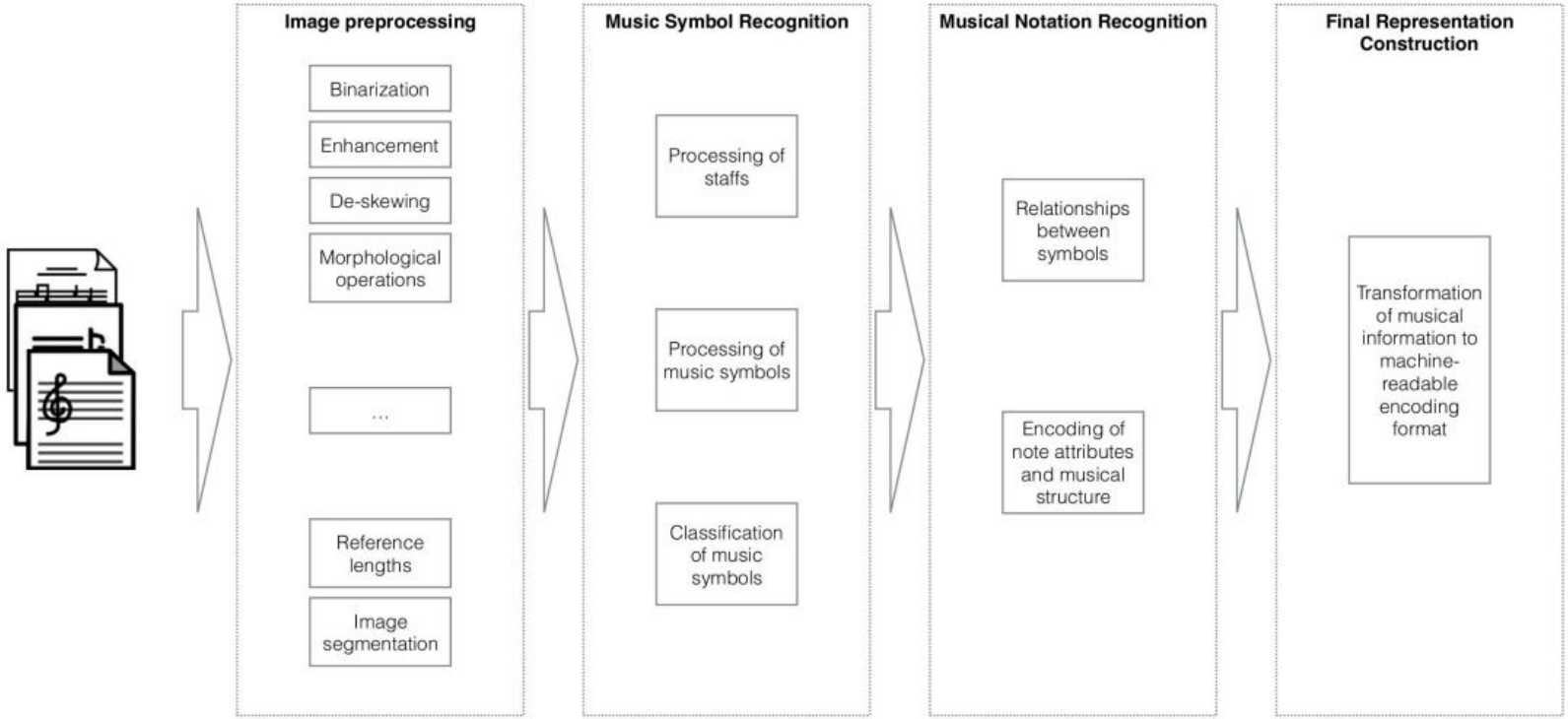
- Preprocessing
- Music Symbol Recognition / Music Object Detection
- Semantical Reconstruction / Notation Assembly
- Encoding

The OMR pipeline



Reproduction of Rebelo et al., 2012

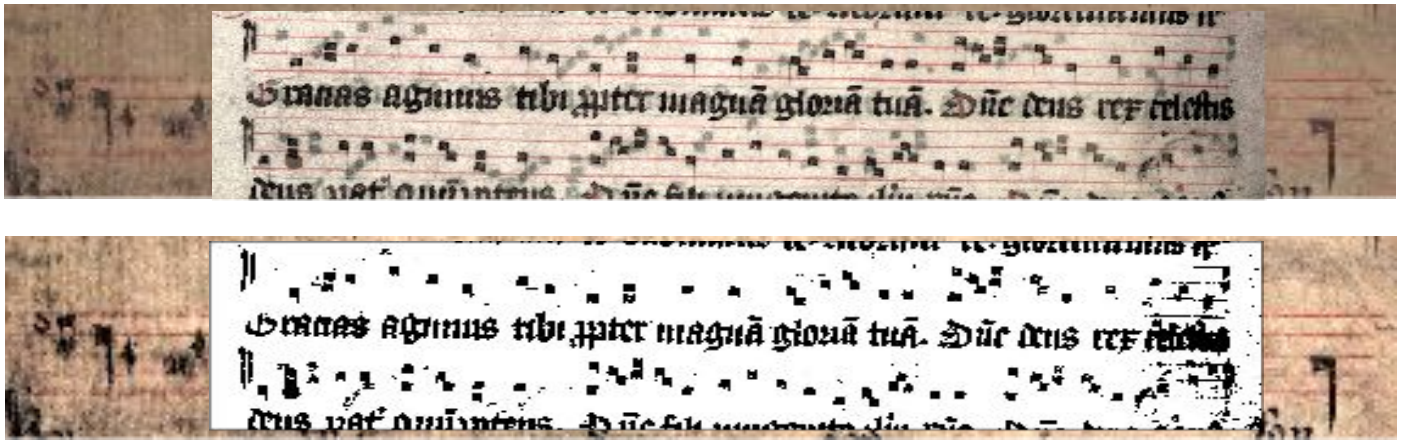
The OMR pipeline



Vigliensoni et al., 2018

Preprocessing

- Ease further steps: contrast enhancement, deskewing, …
- Especially relevant for degraded documents

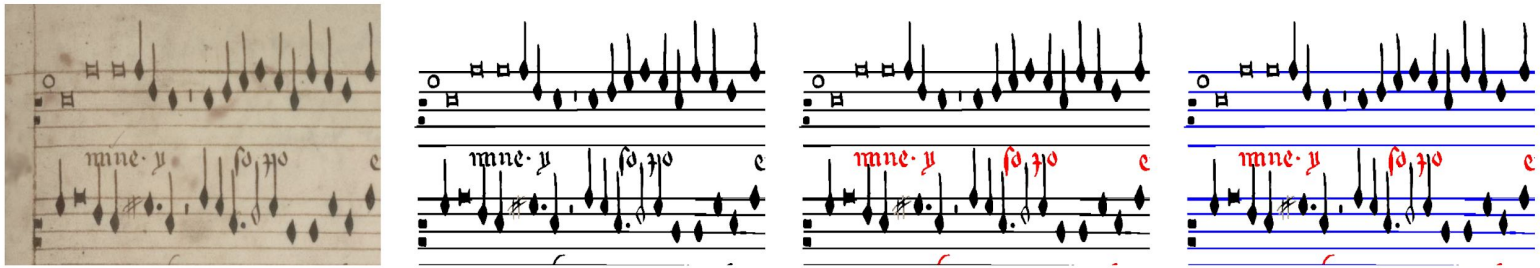


Preprocessing: Document Layout Analysis

- The preprocessing stage must also separate the different layers of *graphical* information
- Traditionally, the layers of interest are:
  - Background
  - Staff lines
  - Music symbols
  - Text
- Comprises several uses
  - Noise removal and enhancement of relevant information
  - Reference scale, deskewing, segmentation of staves
  - Text and music separation
  - Symbol segmentation

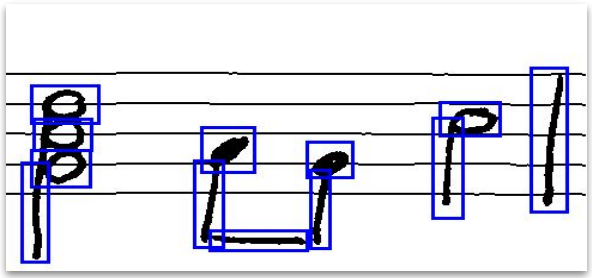
Preprocessing: Document Layout Analysis

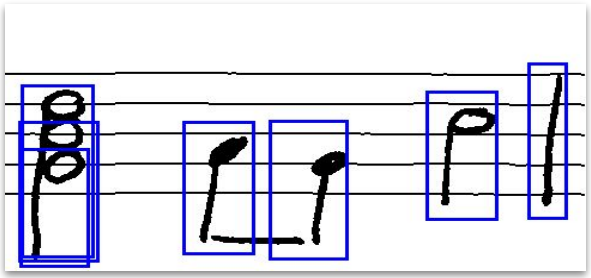
- Number of approaches for dealing with related problems:
  - Binarization
  - Text (lyrics) detection
  - **Staff-line detection**
- Combined use of specific strategies for a complete layout analysis



Music Symbol Recognition

- Detect and categorize music symbol components
- There are two ways of considering the categories
  - Primitives
  - Symbols

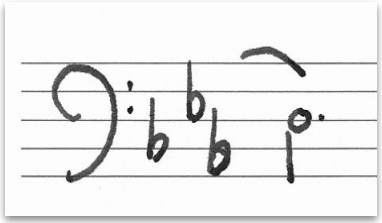




Music Symbol Recognition

Traditional approaches are built upon **symbol segmentation**

- Staff-line removal + connected-component analysis for localization
    - Possible post-process to merge meaningless primitives
- Classification



Music Symbol Recognition

- Staff-line removal stage
  - Traditionally considered the main obstacle for symbol segmentation
  - Much research devoted to solving this stage (Dalitz et al., 2008; Fornés et al, 2013)
  - Currently, it can be included as a part of the layout analysis step

- Classification
  - Comparative study with different classifiers (Rebelo et al., 2012)

Music Symbol Recognition

- Isolating symbols by means of connected components remains problematic
  - Multiple primitives could be connected to each other (e.g., a beam group)
  - A single unit can have multiple disconnected parts (e.g., a fermata, voltas, f-clef)
  - Modelling all possible appearances becomes intractable
    - Especially severe in the context of handwritten notation

Semantical Reconstruction

- Inference of the relationship among the detected symbols
- Two kinds of relations are necessary
  - A **syntactic** relationship (e.g., between a notehead and a stem)
  - A **temporal** relationship to guarantee the correct order of the symbols
- The stage is strongly dependent on how previous steps are formulated
  - Symbol primitives are not strictly unambiguous

Semantical Reconstruction

- Heuristic approaches:
  - Algorithmic rules (Prerau, 1970)
  - Hand-crafted context-free grammars (Fujinaga, 1988; Szwoch, 2007)
  - Generic document recognition based on grammatical formalisms (Couasnon, 2001)
  - Rule-based system with fuzzy modeling (Rossant and Bloch, 2007)
  - Top-down modeling (Raphael and Wang, 2011)

Encoding

- Encodes the recognition into an output format unambiguously
  - MusicXML/MNX/MEI
  - MIDI
- Little attention received from the OMR literature
  - Within OMR, it can be considered an out-of-scope research topic
  - Other communities are doing actual research on this matter.
  - Choice of output encoding, however, impacts choice of intermediate representations

- **Lecturer’s opinion:** this should not belong to OMR.

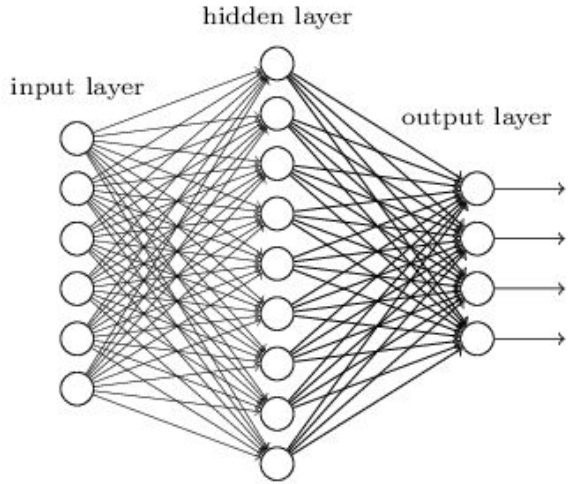
State of the art: Deep Learning

- Deep Learning (DL) has dramatically changed the machine learning paradigm by providing powerful models for many disparate duties, especially in computer vision.
- DL refers to the use of the new generation of Artificial Neural Networks: Deep Neural Networks

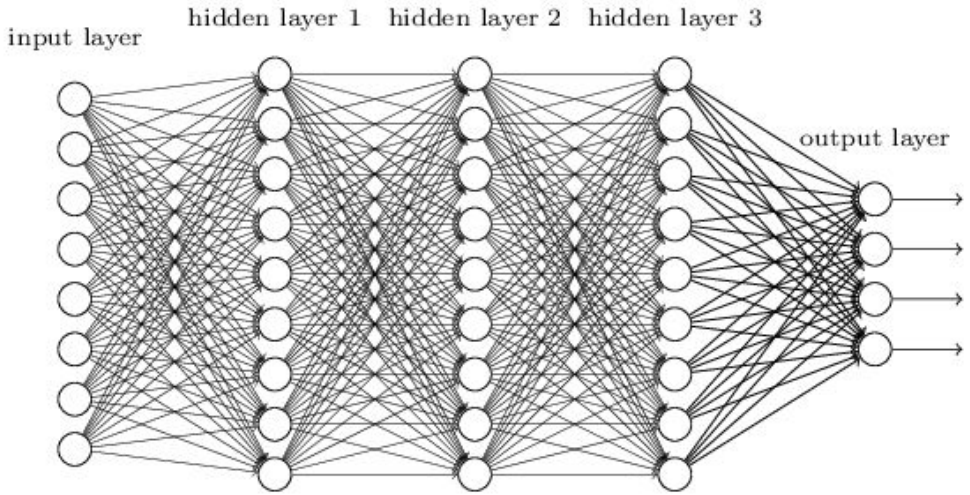
Deep Learning

- New algorithms for training and regularising neural networks, as well as new types of neurons that allow deeper networks

"Non-deep" feedforward neural network

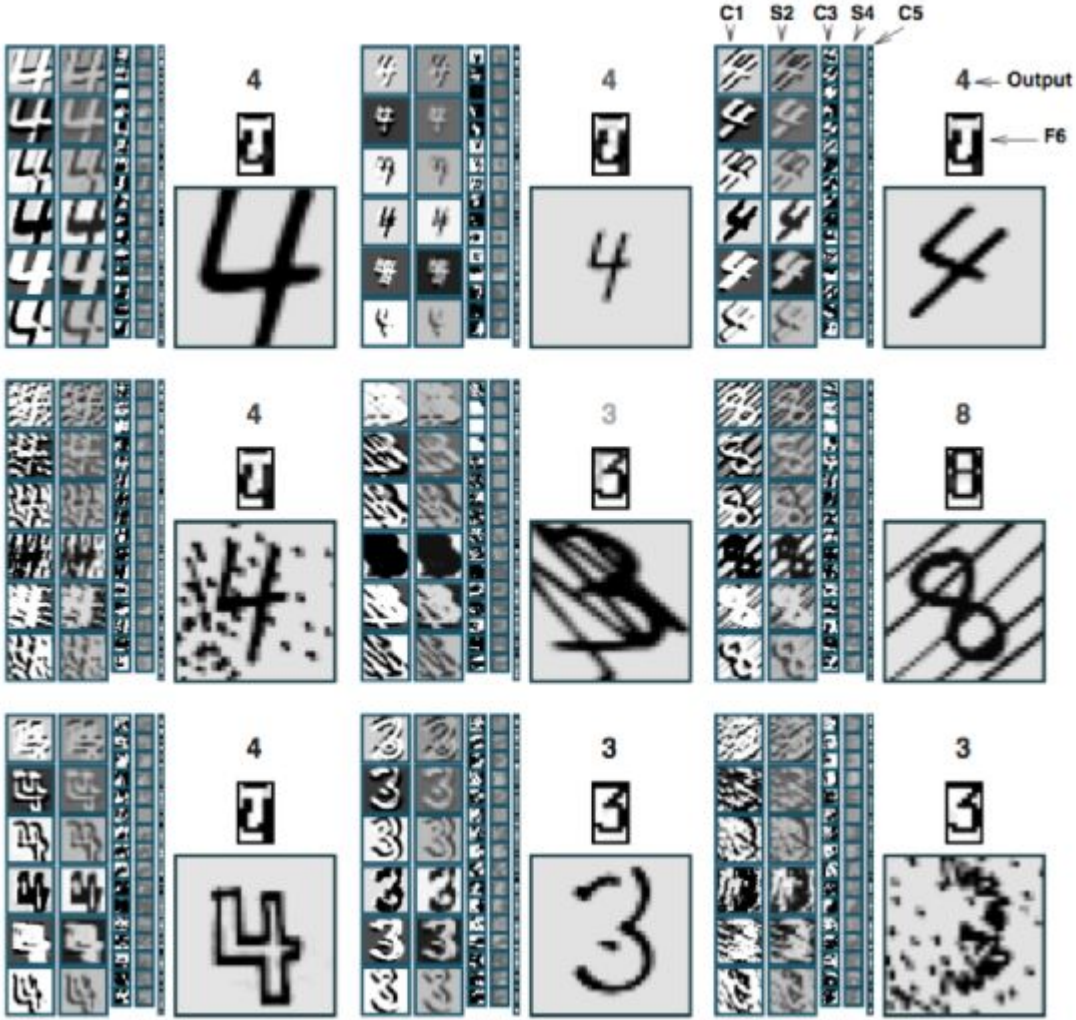


Deep neural network



Deep Lea

- Handcraft ierarchical feature ex
  - These epresentation of the tas
  - Their le



Deep Learning

- *Koray Kavukcuoglu* from DeepMind:
  - “Perhaps the biggest factor [behind recent advances in deep learning] has been the huge increase of computational power. This has made it possible to train much larger and deeper architectures, yielding dramatic improvements in performance. More is more when it comes to neural networks.”
  - “Another catalyst has been the availability of large labelled datasets for tasks such as speech recognition and image classification.”
  - “At the same time our understanding of how neural networks function has deepened, leading to advances in architectures, optimisation algorithms, and regularisation.”

OMR in the era of Deep Learning

- With the appearance of DL in OMR, many steps that traditionally produced suboptimal results have seen drastic improvements
  - For instance, symbol classification or staff-line removal.
- This also caused some steps to become obsolete or collapse into a single (bigger) stage.
  - Deep learning models have been shown to be able to deal with **direct music object detection** stage, without having to remove staff lines at all.

OMR in the era of deep learning

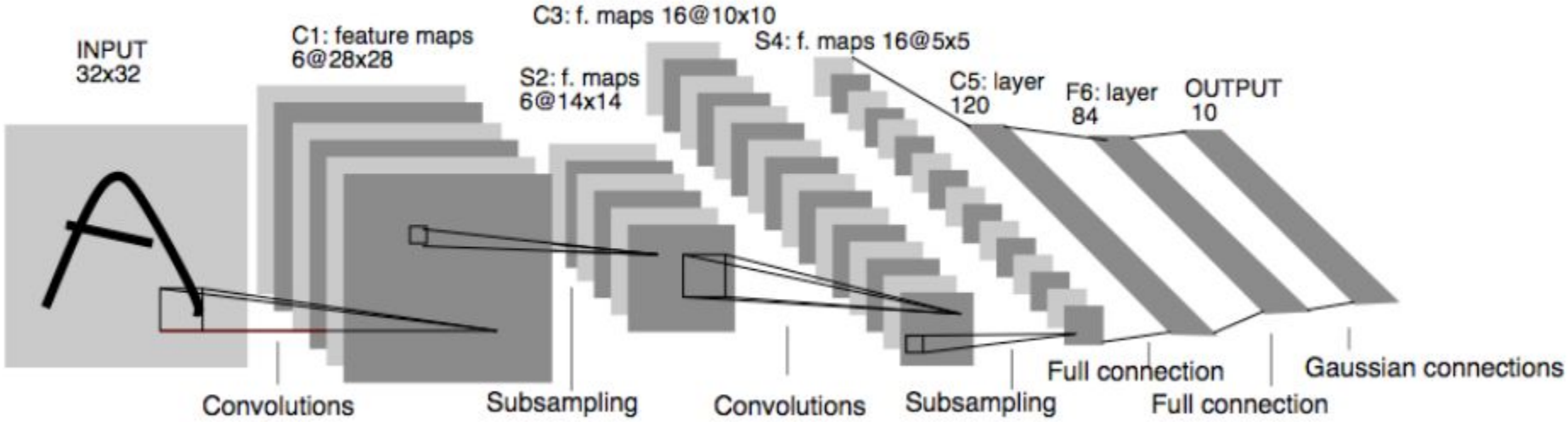
- These recent advances also diversified the way of how OMR is approached altogether: there are alternative pipelines with their own ongoing research that attempt to face the whole process in a single step.
- This holistic paradigm, also referred to as **end-to-end systems**, has been dominating the current state of the art in other tasks such as text, speech, or mathematical formula recognition.

Deep Learning in the OMR pipeline: Convolutional Neural Networks

DL in OMR: Convolutional Neural Networks

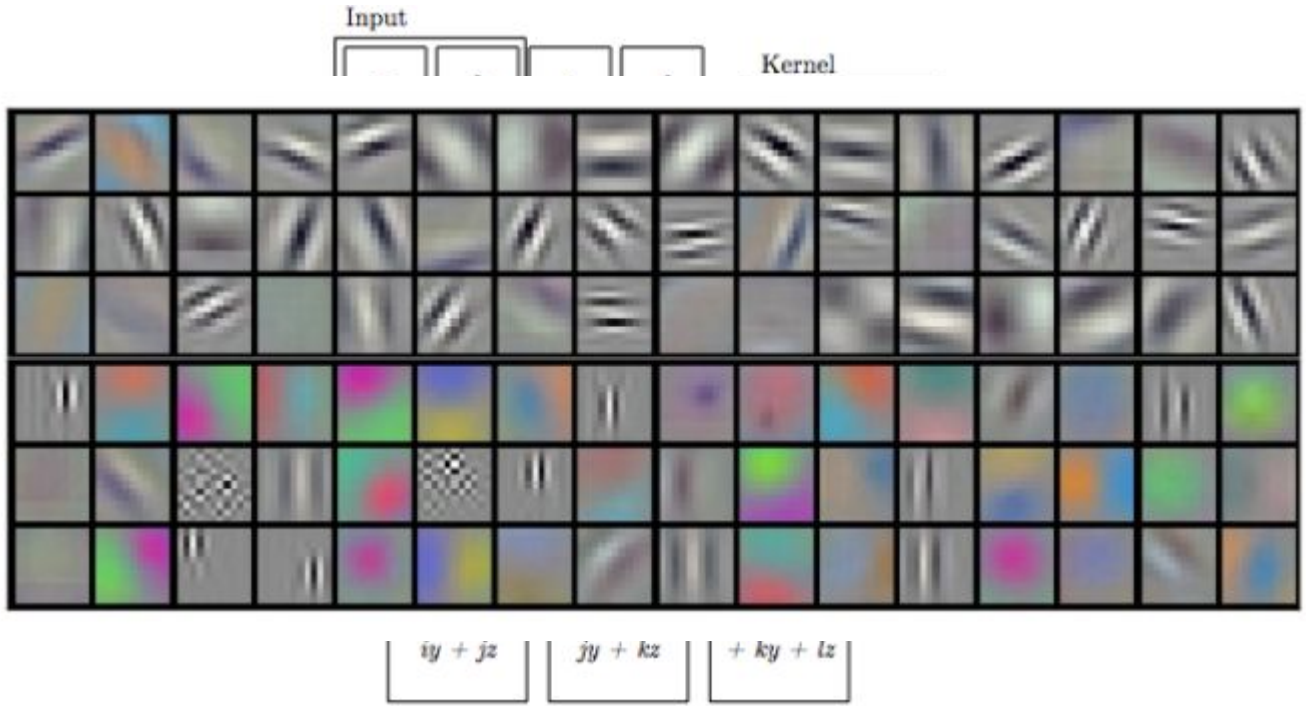
- OMR is primarily a computer vision challenge
- The DL models that had a higher impact in the computer vision community are **Convolutional Neural Networks** (CNN)
  - State of the art in computer vision and image processing tasks
  - Hierarchy of filters (convolutions) that process an image to solve some task
  - Filters are not fixed but **learned through a training process**
  - Feature extraction is no longer necessary
  - A labeled training set (ground-truth data) is required

Convolutional Neural Networks



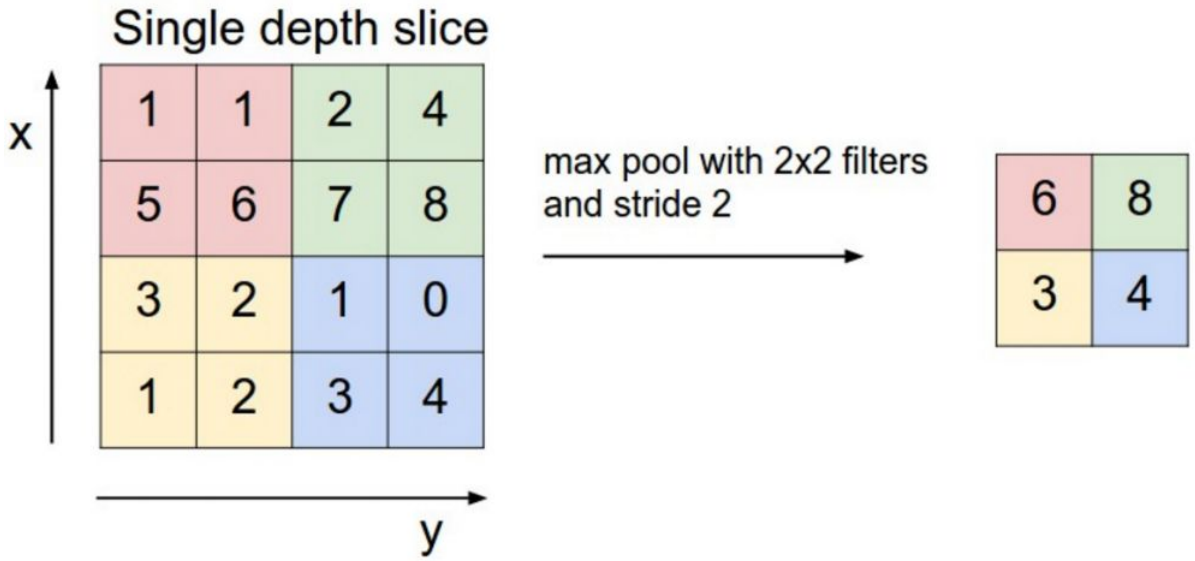
Convolutional Neural Networks

Convolution



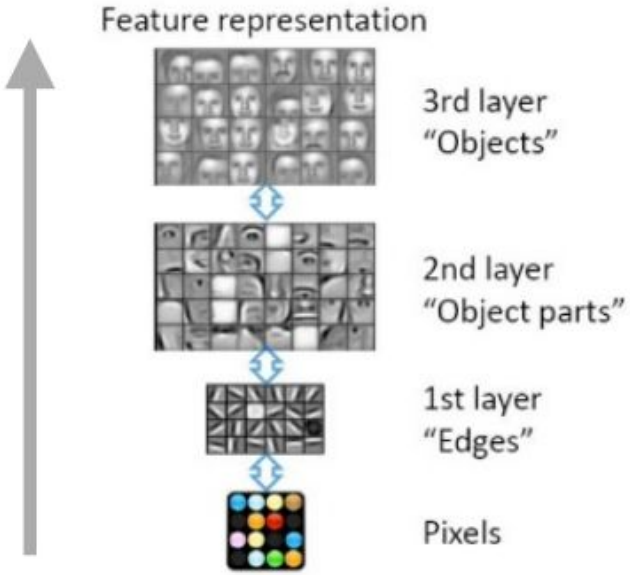
Convolutional Neural Networks

Subsampling: max-pooling



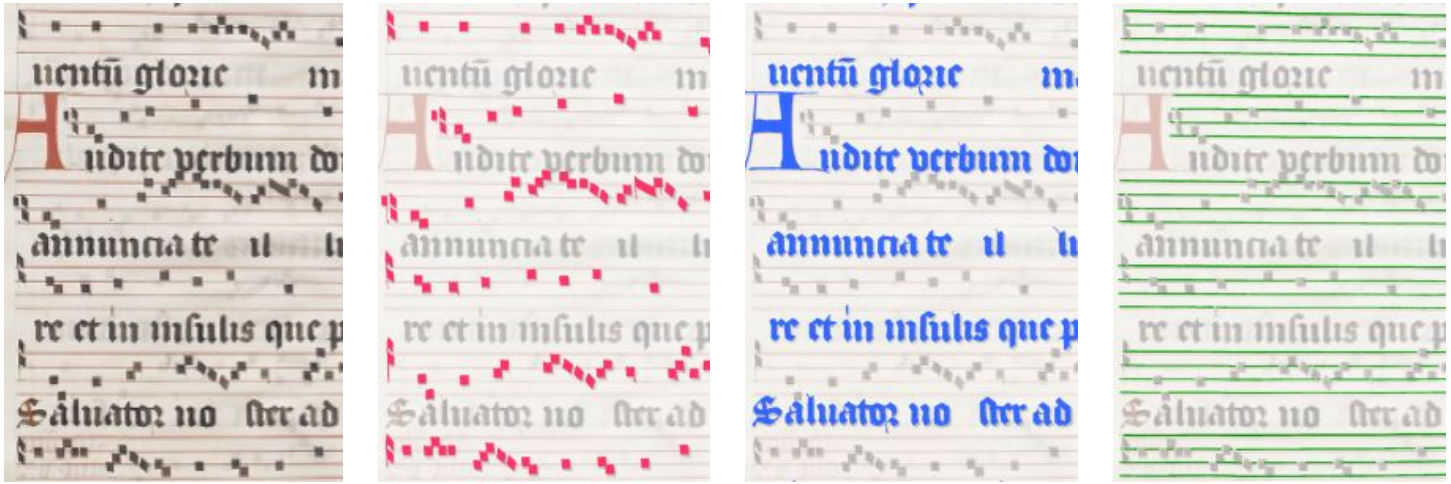
Convolutional Neural Networks

Hierarchical feature extraction



DL in OMR: Convolutional Neural Networks

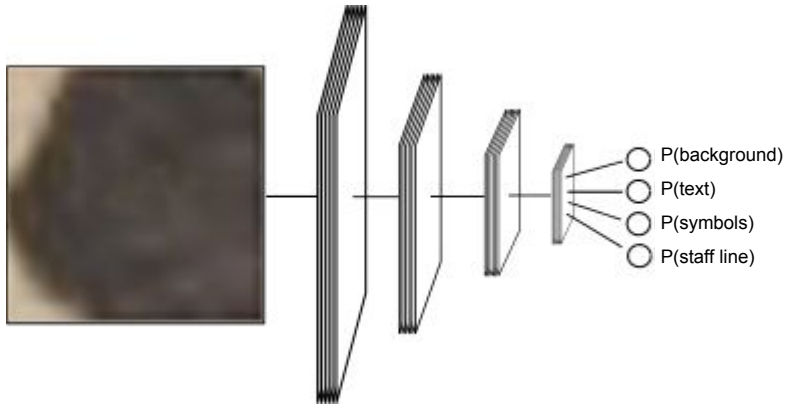
- Direct layout analysis with CNN
  - Formulate the problem as a pixel-wise classification task
  - Solve the task with machine learning strategies
  - Ground-truth data is required



Pixel-level ground-truth

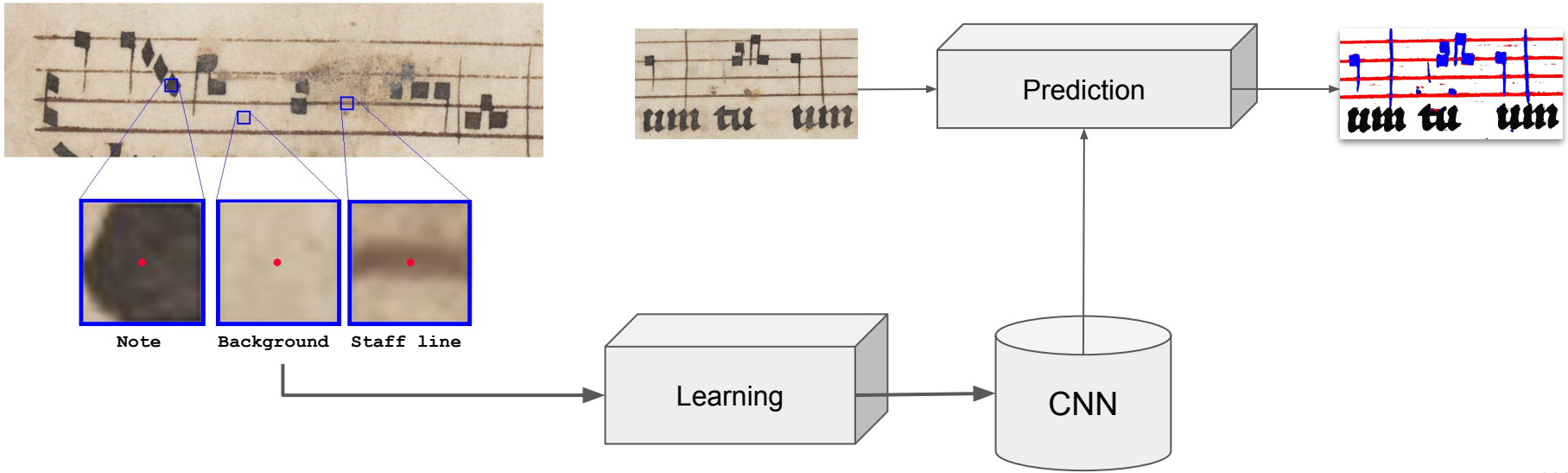
DL in OMR: Convolutional Neural Networks

Pixel-wise classification with Convolutional Neural Networks
(Calvo-Zaragoza et al., 2018)



DL in OMR: Convolutional Neural Networks

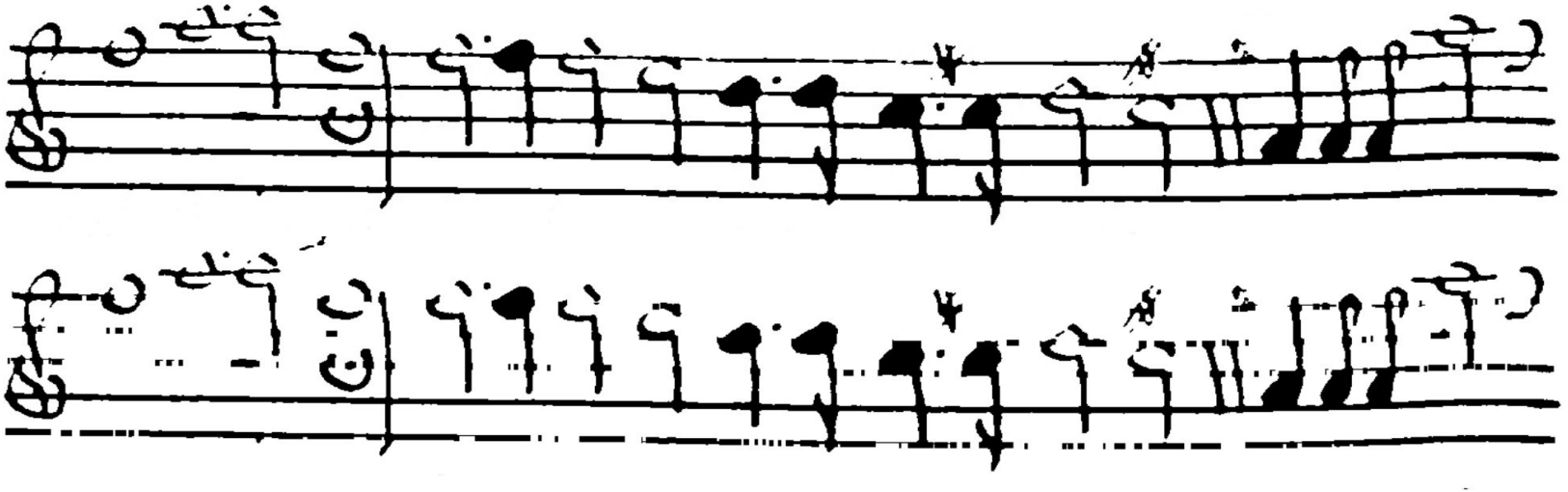
Pixel-wise classification with Convolutional Neural Networks
(Calvo-Zaragoza et al., 2018)



DL in OMR: Convolutional Neural Networks

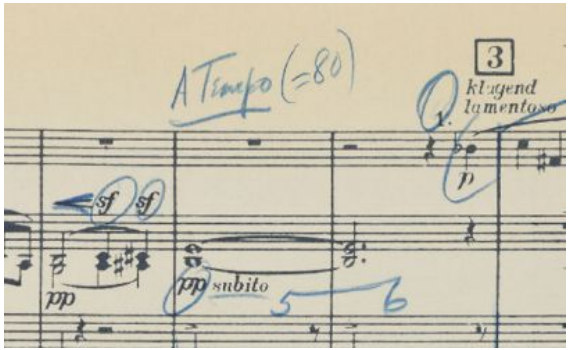
How was the situation before using DL?

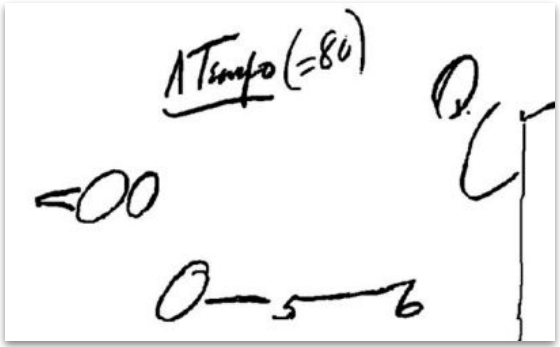
- Performance of the winner of the latest ‘Contest on staff-line removal’

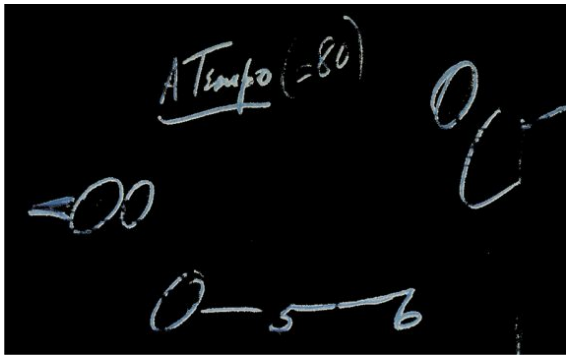


DL in OMR: Convolutional Neural Networks

Extended to handwritten annotation extraction (Bell and Pugin, 2018)

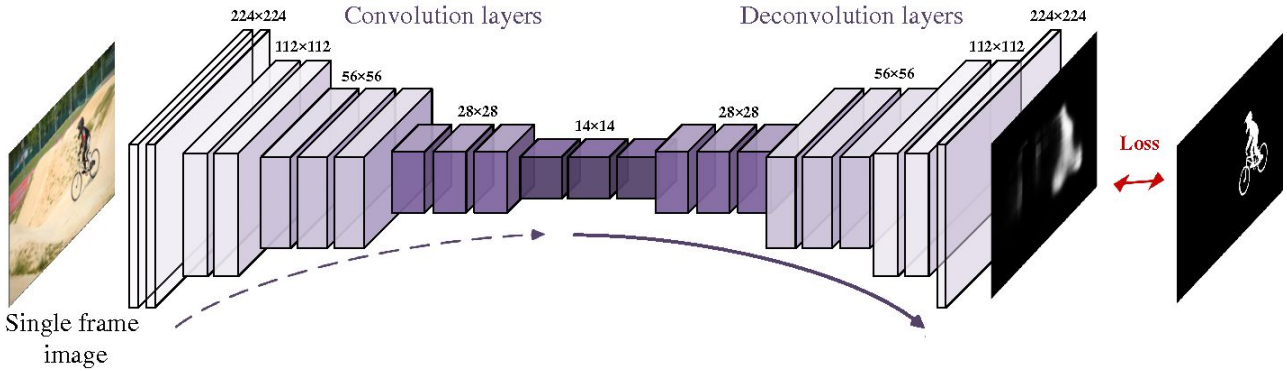






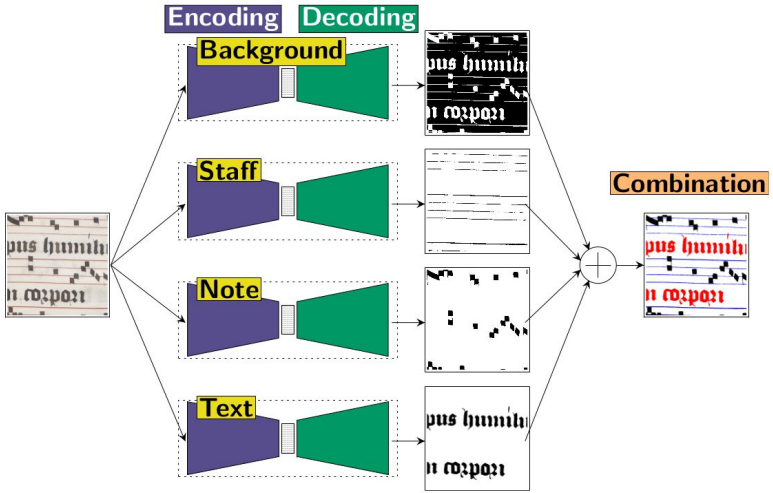
Fully Convolutional Neural Networks

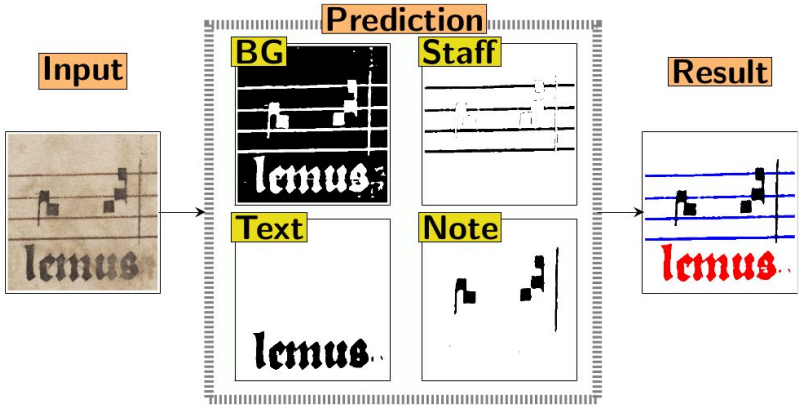
- Given that CNN are composed by filters that produce a new image, we can learn an image-to-image neural network.
- This is called fully-convolutional neural networks.



DL in OMR: Convolutional Neural Networks

Patchwise approach with Fully-Convolutional Neural Networks
(Castellanos et al., 2018)

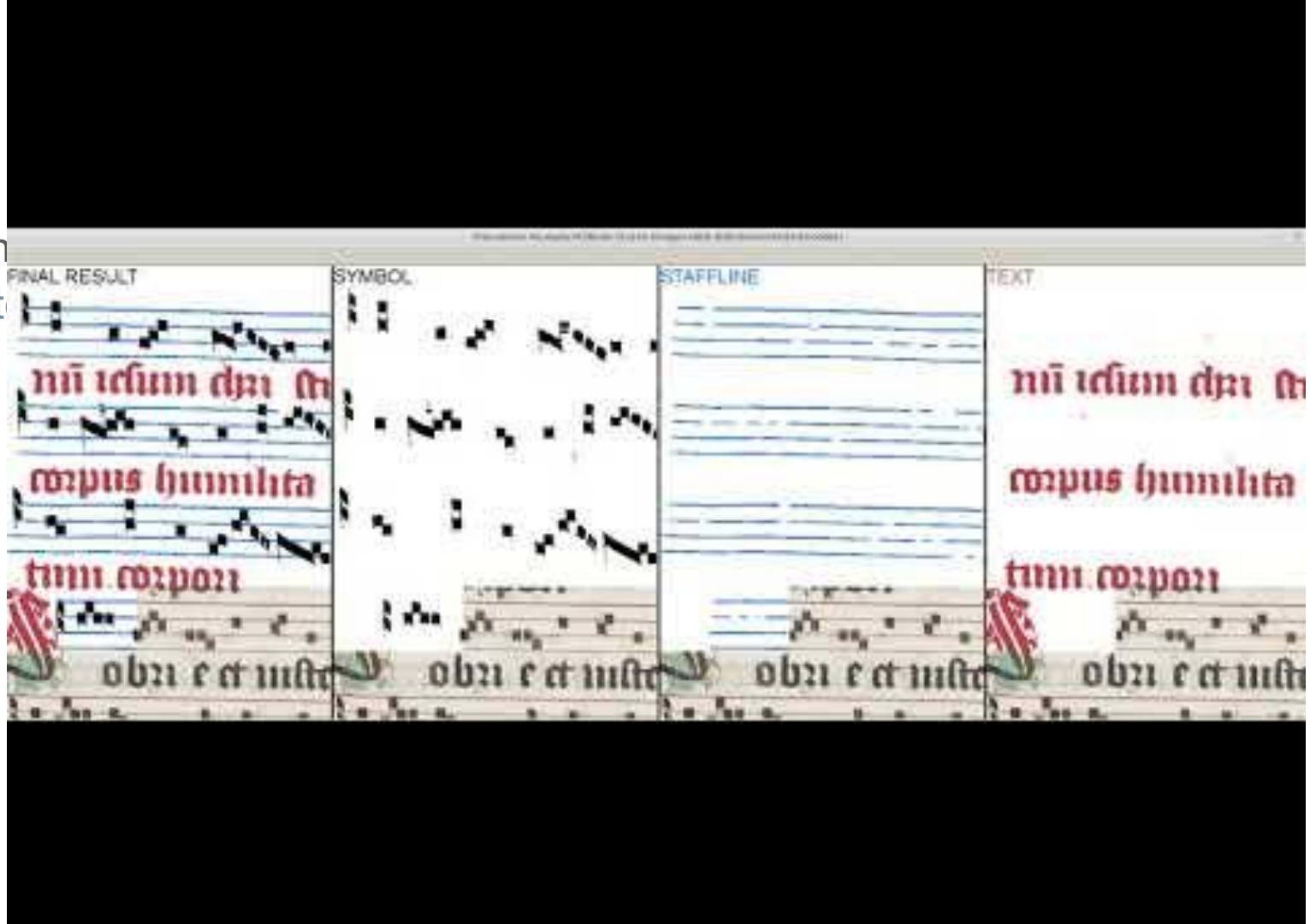




DL

Patch

(Cast



DL in OMR: Convolutional Neural Networks

Comparison between CNN-based approaches for Document Layout Analysis:

Model	Accuracy	Training time	Processing time	Ground-truth
Patchwise	~ 92 %	~ 5 hours	~ 1 minute per page	~ 5 full pages
Pixelwise	~ 90 %	~ 5 hours	~ 6 hours per page	~ 1 full page

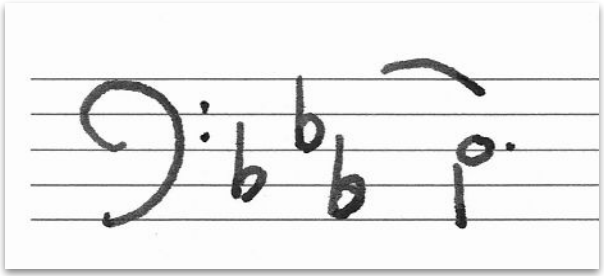
DL in OMR: Convolutional Neural Networks

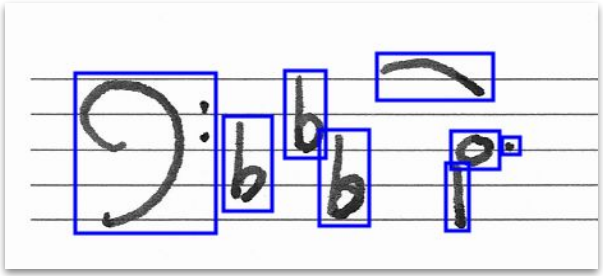
- **Isolated music symbol classification** is no longer a challenge.
- CNN are able to classify any well-defined set of music-symbol categories (Pinheiro et al., 2016; Lee et al., 2016; Calvo-Zaragoza et al, 2017; Pacha and Eidenberger, 2017)
- As long as the symbols **have been previously isolated** and we have a proper training set of such symbols

Music Object Detection

Currently, there is a trend towards **direct music-object detection**

- Detection: localization + classification
- Advantages
  - Elegant formulation: from image to set of bounding boxes
  - **Predefined set of primitives** directly considered
  - Single training stage

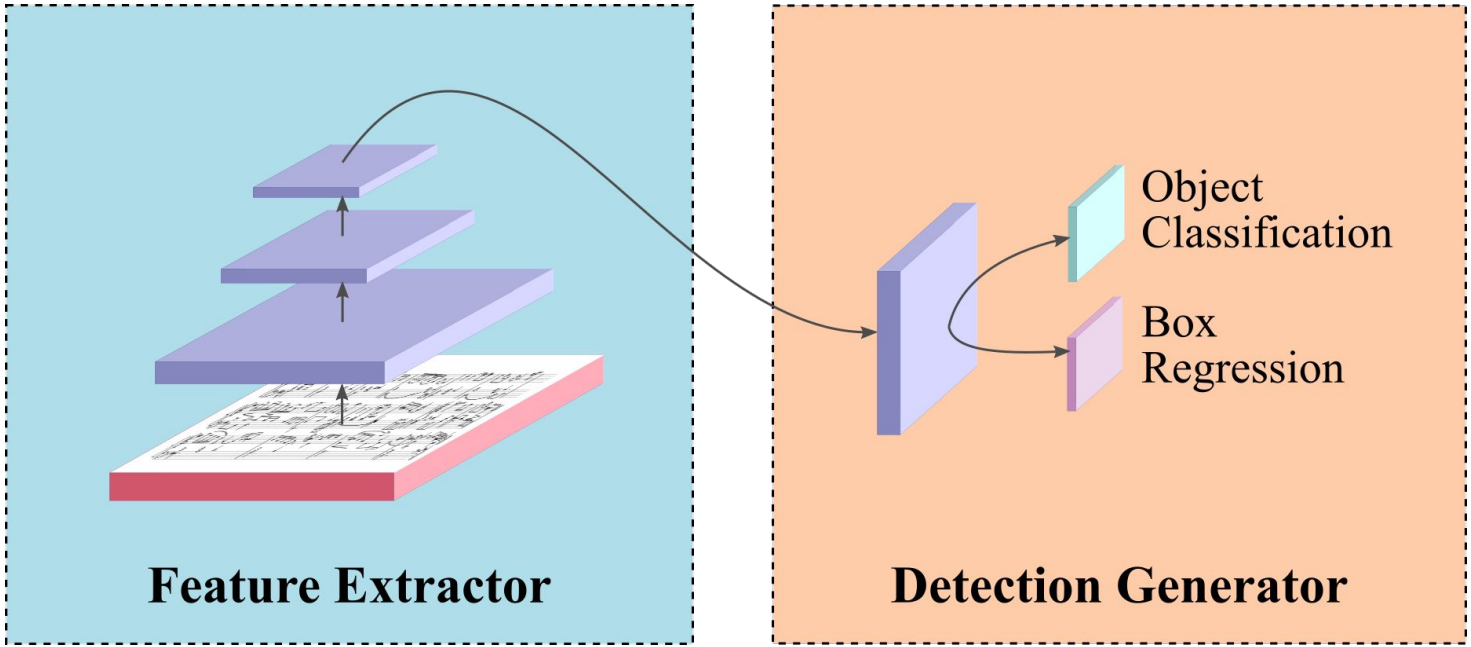




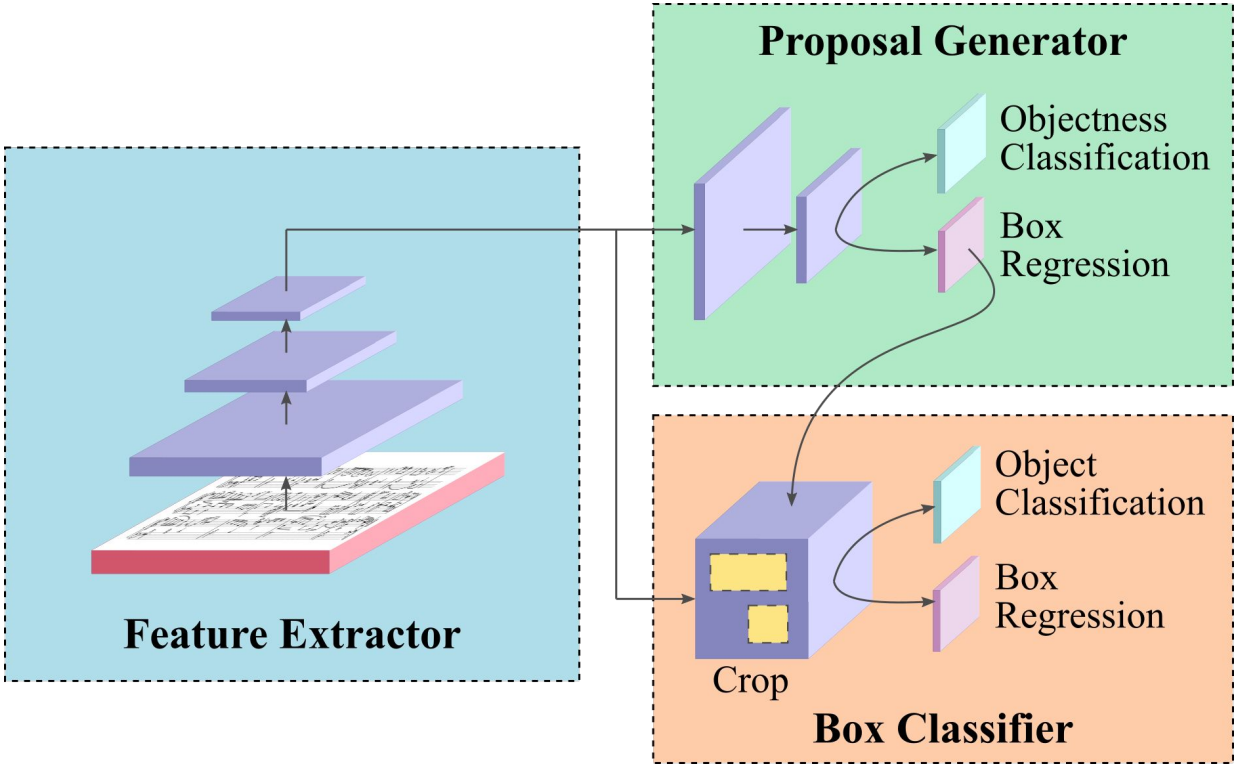
Music Object Detection

- State-of-the-art models
  - General object detectors from the Computer Vision community
    - Models used for general object detection can be applied for music object detection (Pacha et al., 2018)
    - There are two main approaches
      - One-stage detector
      - Two-stage detector
  - U-Nets (Hajič jr., 2018a)
  - Deep Watershed Detector (Tuggener et al., 2018)

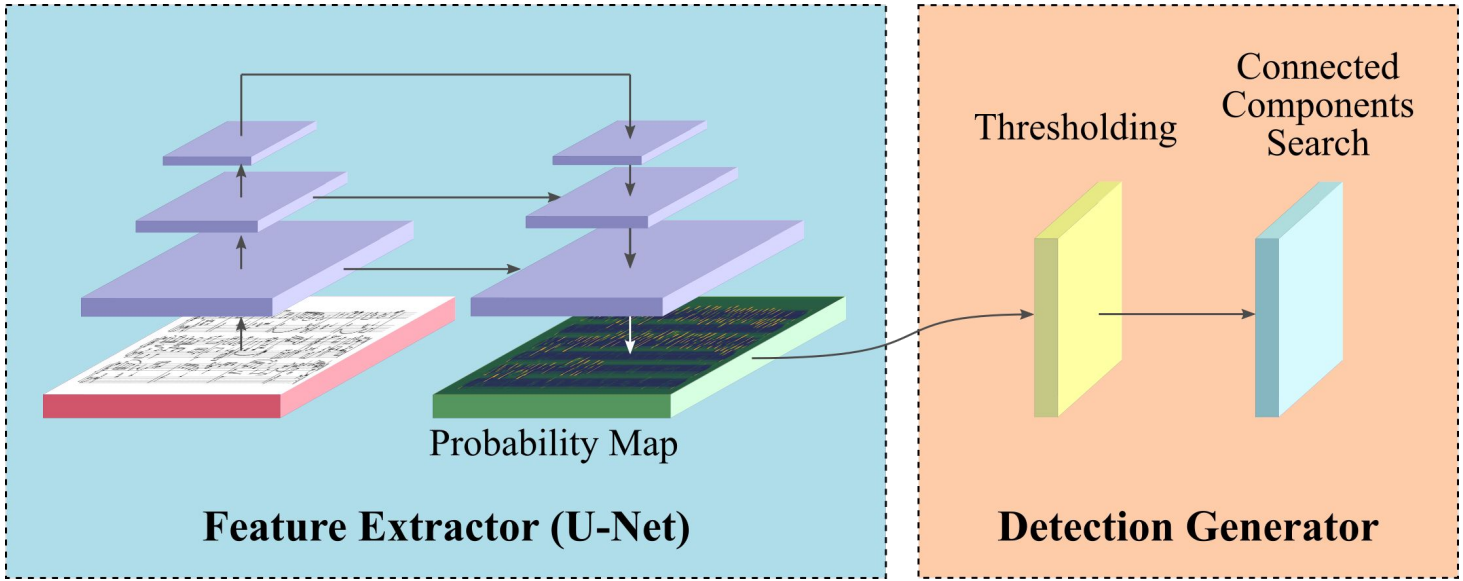
One Stage Detector



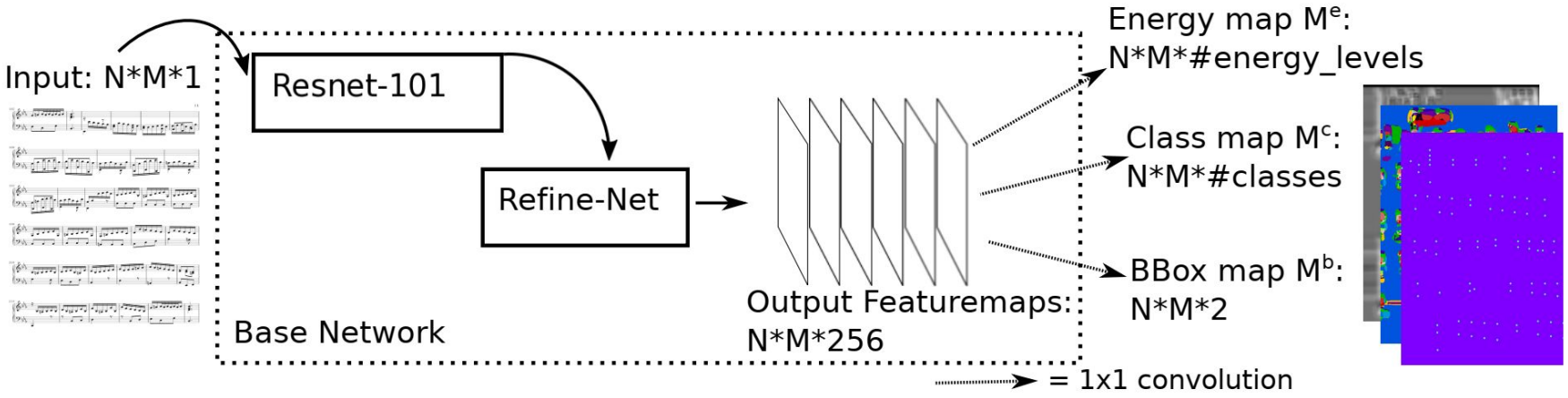
Two Stage Detector

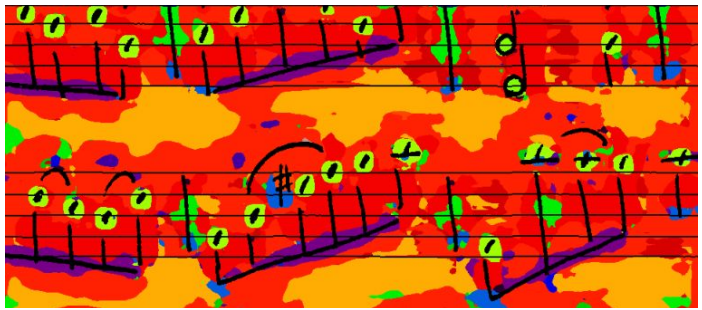


U-Net Detector



Deep Watershed Detector





Music Object Detection - Comparison (2018)

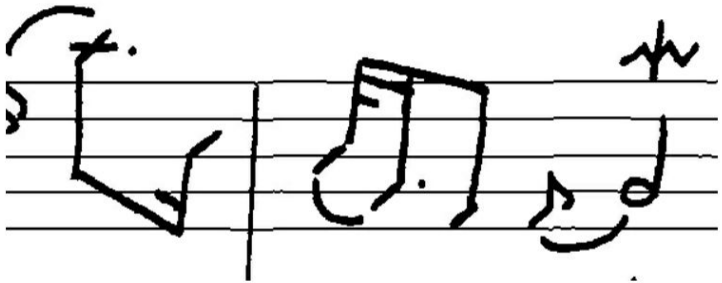
	One stage detectors	Two stage detectors	U-Net	DWD*
Accuracy (Printed CWMN)	9.8	19.6	24.8	< 16.0
Accuracy (Handwritten CWMN)	7.7	3.9	16.6	*unknown*
Accuracy (Handwritten Mensural)	14.5	15.2	17.4	*unknown*
Training time (per dataset)	1-2 h	8-12 h	3+ days	3+ days
Prediction time (per page)	1 s	20-50 s	40-80 s	3-5 s

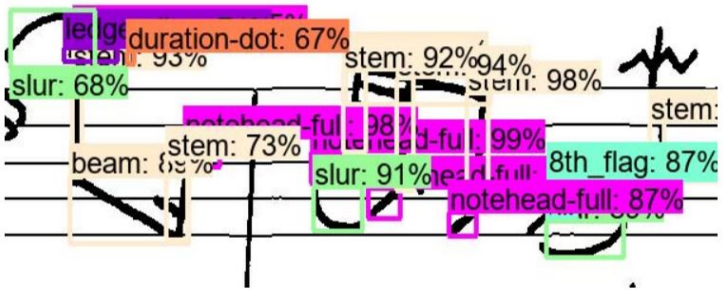
* Approximated from non-comparable experiments.

Music Object Detection

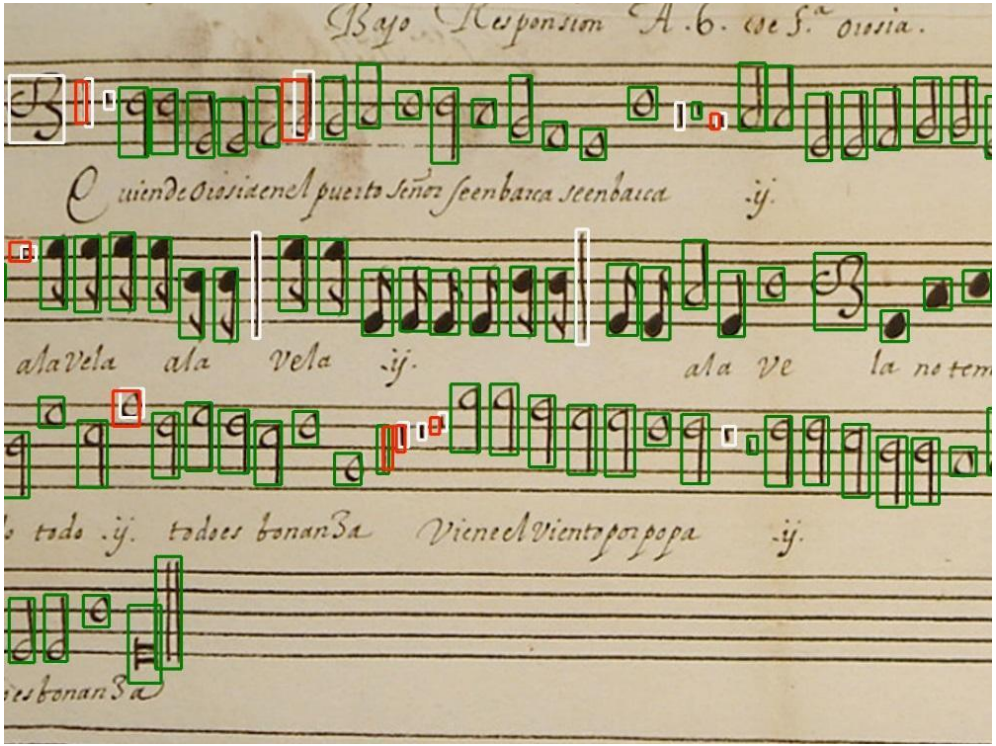
- Previous results were too pessimistic
  - The performance can be greatly improved
    - New general object detectors from the computer vision community
    - Staff-splitting is significatively beneficial
  - Standard object-detection metrics are not adequate for OMR

Music Object Detection



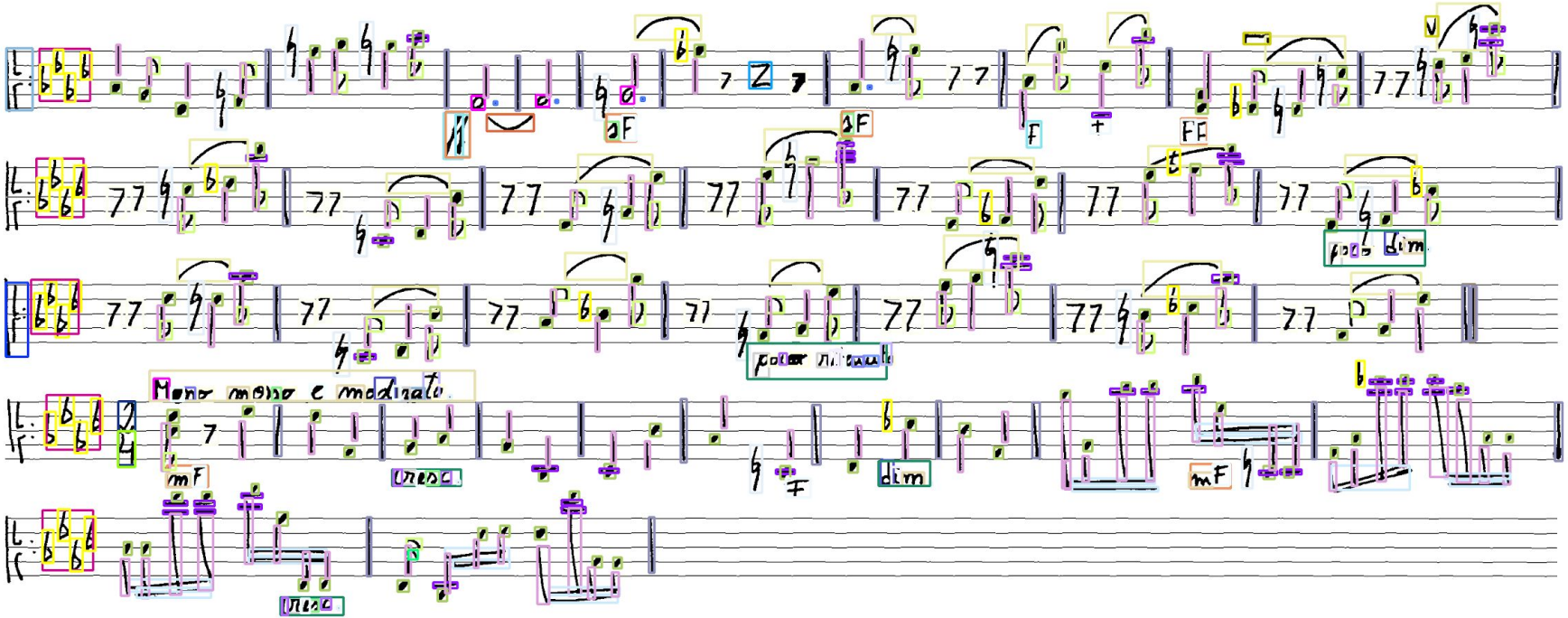


Music Object Detection



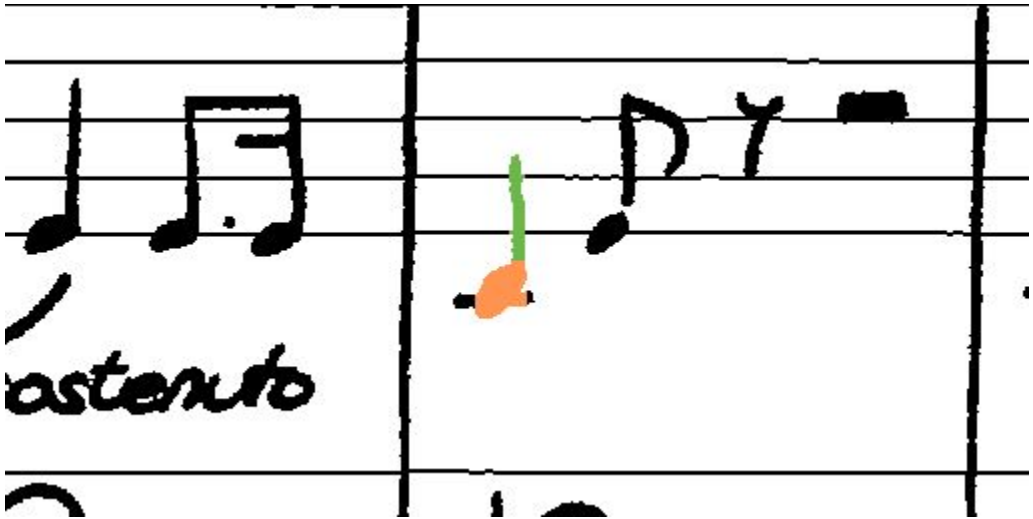
Recent advances (2019)

- Complete OMR pipeline with DL approaches
  - Music-object detection + DL-based Notation assembly stage (unpublished):
    - Once primitives have been detected, a binary DL classifier is trained that must predict whether two objects are related or not.
    - It receives a 3-channel image with
      - Channel 1: the crop of the image that covers both primitives
      - Channel 2: the crop of the image with a mask over object 1
      - Channel 3: the crop of the image with a mask over object 2
    - Incompatible primitives are directly discarded:
      - Primitives that are far apart in the image
      - Primitives that cannot be syntactically related


Meno mosso e moderato
mF
dim
mF
dim

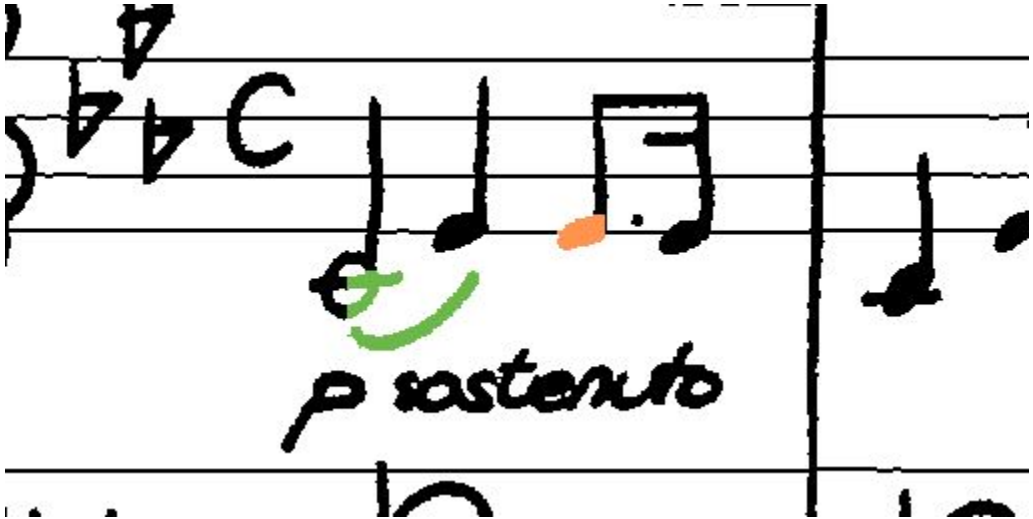
Recent advances (2019)

- Complete OMR pipeline with DL approaches
  - Music-object detection + DL-based Notation assembly stage (unpublished):



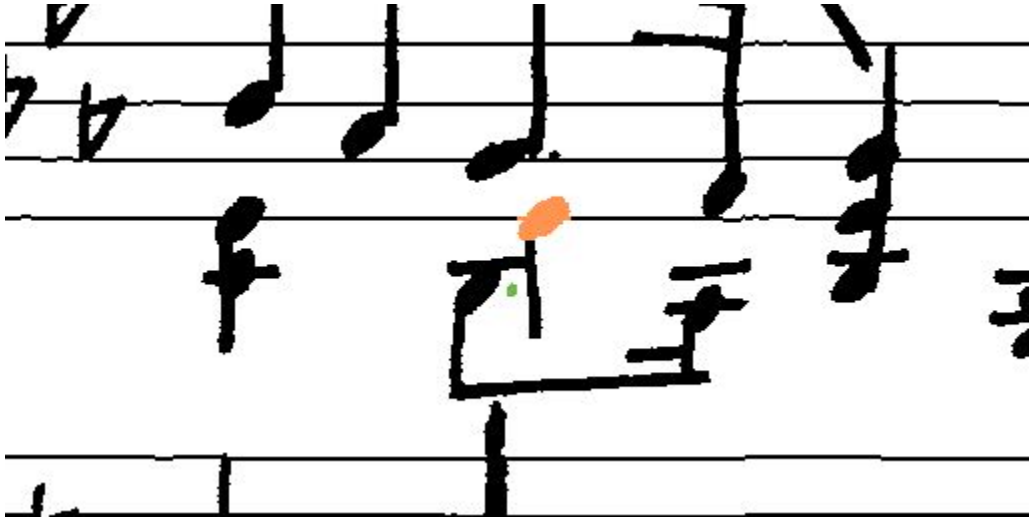
Recent advances (2019)

- Complete OMR pipeline with DL approaches
  - Music-object detection + DL-based Notation assembly stage (unpublished):



Recent advances (2019)

- Complete OMR pipeline with DL approaches
  - Music-object detection + DL-based Notation assembly stage (unpublished):



Recent advances (2019)

- Complete OMR pipeline with DL approaches
  - Music-object detection + DL-based Notation assembly stage (unpublished):

	Graph Edges / Relationships		
	Precision	**Recall**	**F-Score**
Perfect Detection	95.2%	96.0%	95.6%
Real Detector	93.2%	91.5%	92.3%

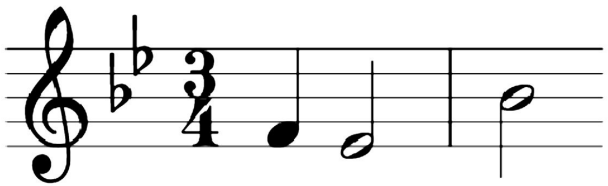
Deep Learning for end-to-end OMR: Recurrent Neural Networks

End-to-end OMR

- Formulate the task without any further subdivision
- Sequence of musical symbols in the score is directly obtained
- Advantages:
  - Holistic process - the score is seen as a whole, without paying explicit attention to its constituent components
  - Less demanding ground-truth data (no geometrical information is necessary)

End-to-end OMR

- There is no model for end-to-end recognition of full music score images
  - Because of design limitations, current models only deal with tasks whose output can be expressed as a sequence
  - However, this is enough for single-staff end-to-end recognition

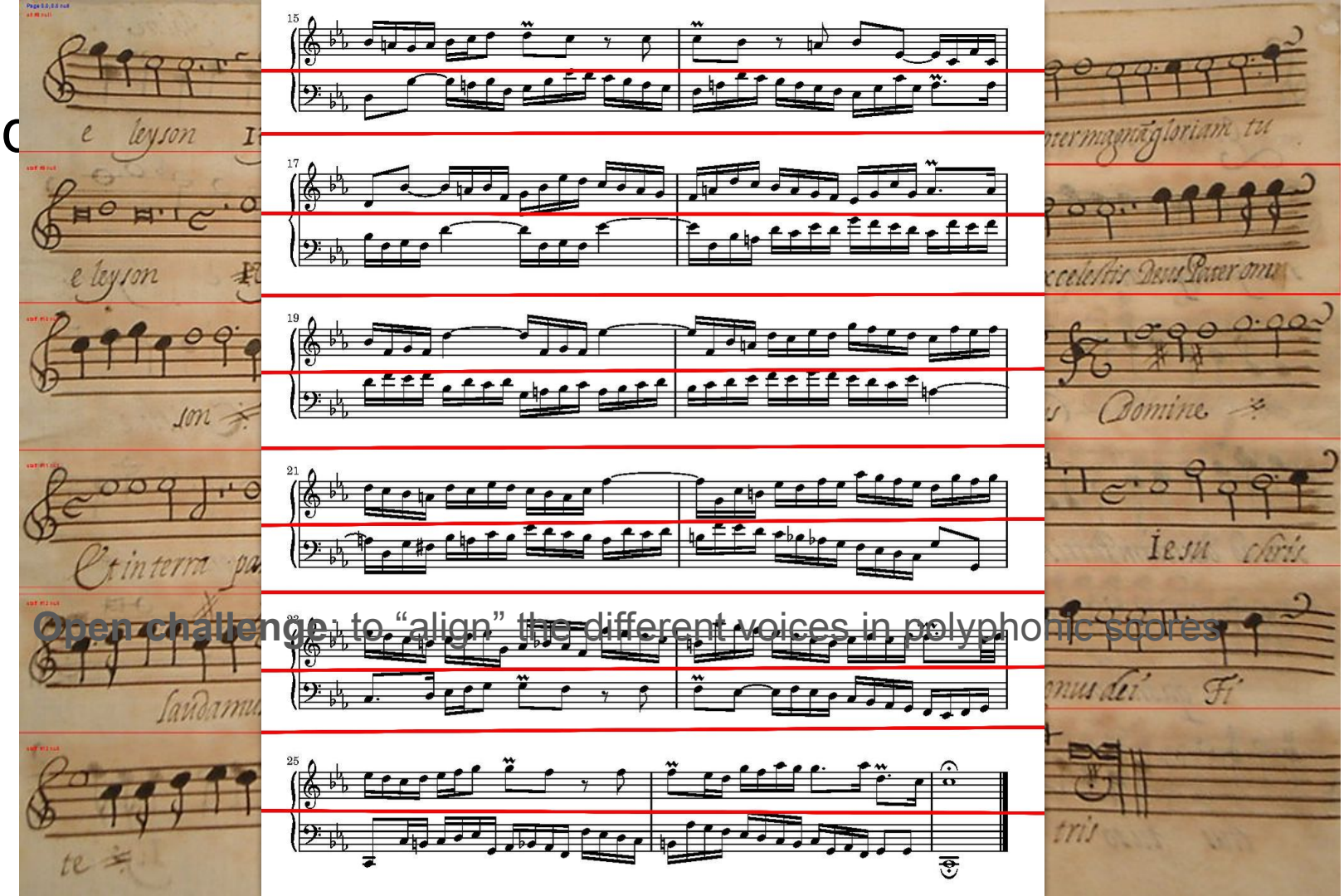


G4 Clef
B ♭ Key
¾ Time Signature
F4 Quarter Note
E4 Half Note
Barline
B ♭ 4 Half Note

Enc

-

- **Open challenge**: to “align” the different voices in polyphonic scores

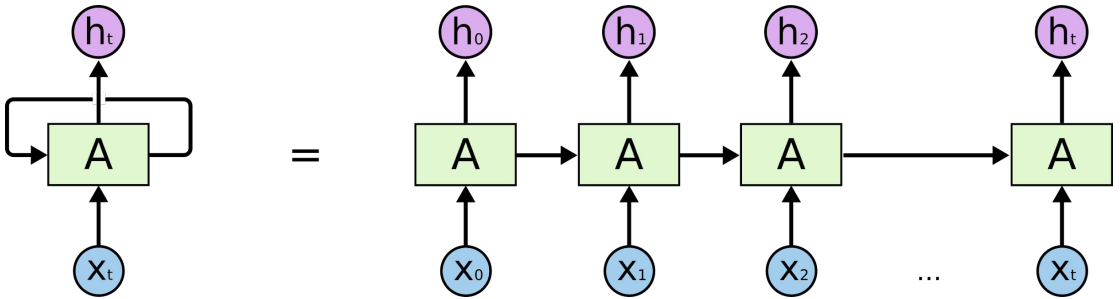


End-to-end OMR

- Pioneered by Laurent Pugin using hidden Markov models (Pugin, 2006)
- Recent approaches with Deep Learning:
  - Encoder-Decoder Convolutional Recurrent Neural Networks (van der Wel & Ullrich, 2017)
  - Convolutional Recurrent Neural Network + Connectionist Temporal Classification (Calvo-Zaragoza et al., 2017; Calvo-Zaragoza & Rizo, 2018)

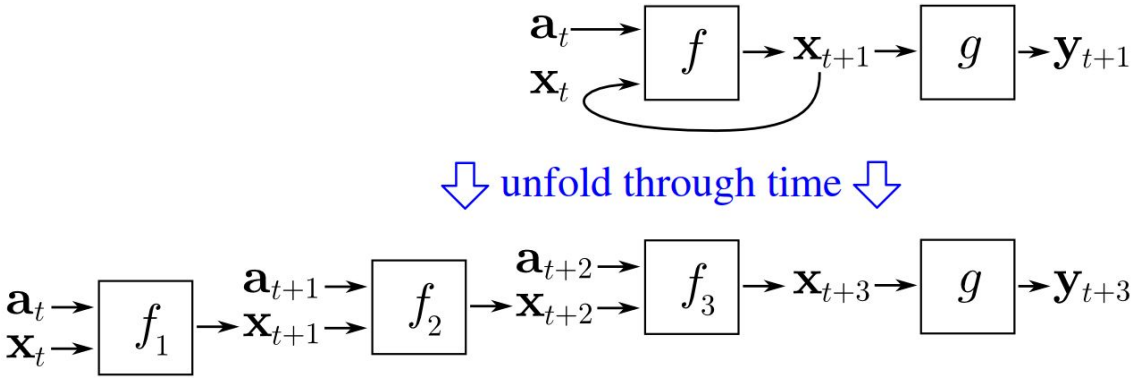
Recurrent Neural Networks

- CNN are not suitable to handle sequential data because they only condition the output to the current input
- Recurrent Neural Networks (RNN) have a hidden state that is propagated through time:
  - Every output is conditioned both by the input and the last state of the network



Recurrent Neural Networks

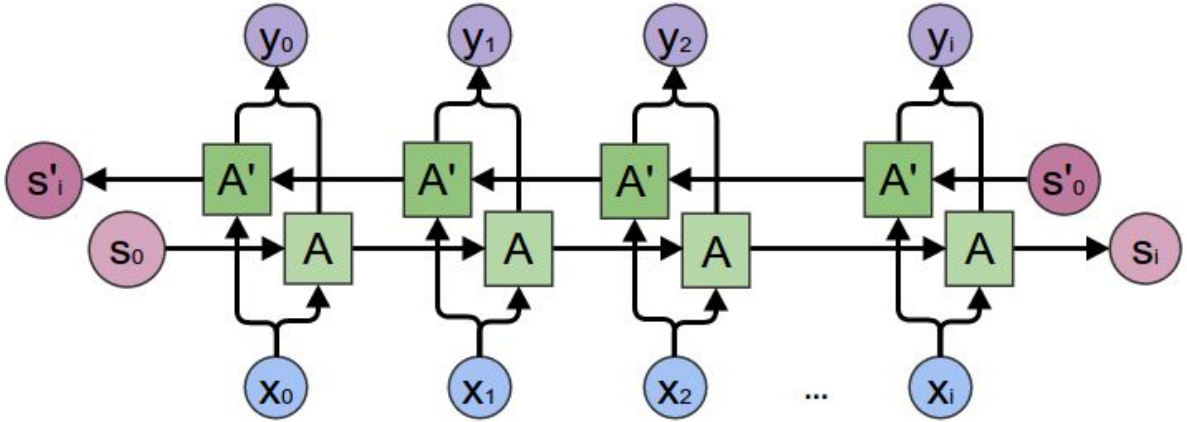
- The learning process follows the same principles



- Training RNNs is more complex computationally

Recurrent Neural Networks

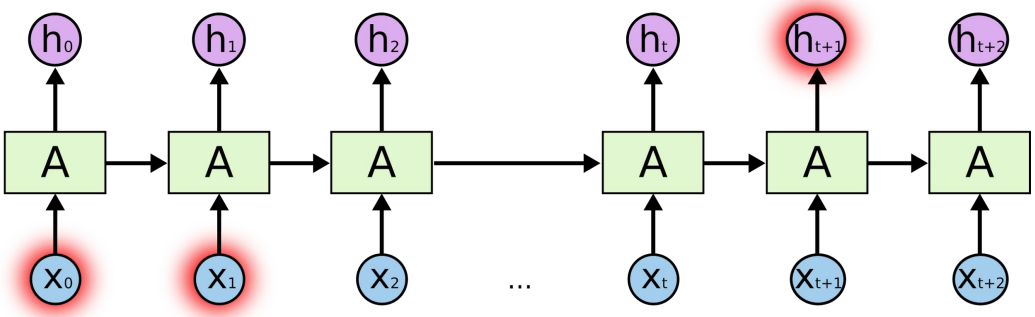
Bidirectional RNN



Recurrent Neural Networks: LSTM

- The RNN state at a particular time only consider short-term events
- Long-term dependencies cannot be dealt with
  - Poor training and generalization
- Solution: Long short-term memory (LSTM) cells
  - Special RNN neuron with a little memory
  - LSTM cells are good at remembering dependences for a long period of time due to its internal architecture
    - They actively learn to remember long-term dependencies

Recurrent Neural Networks: LSTM



Recurrent Neural Networks: LSTM

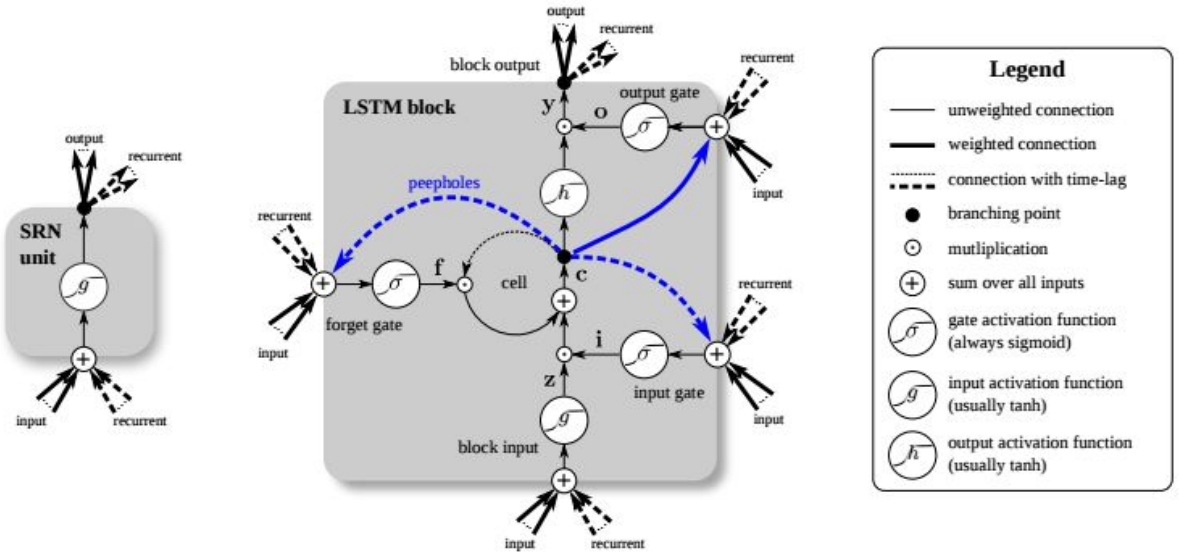


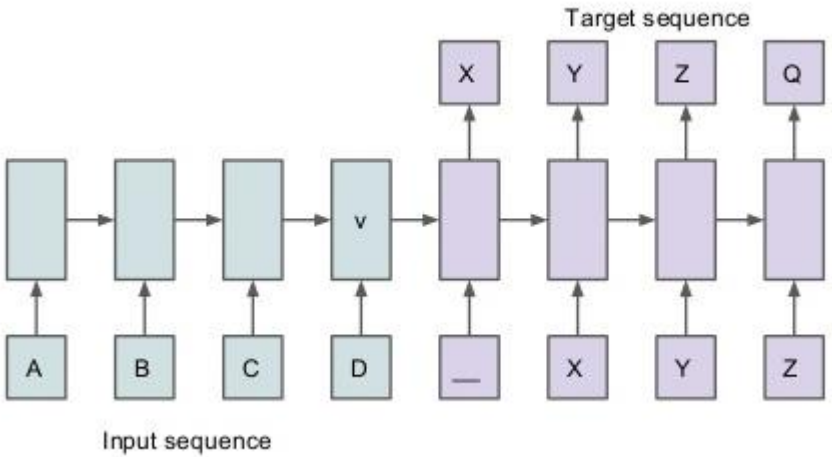
Figure 1. Detailed schematic of the Simple Recurrent Network (SRN) unit (left) and a Long Short-Term Memory block (right) as used in the hidden layers of a recurrent neural network.

Encoder-Decoder Recurrent Neural Network

- A particular RNN architecture is that of encoder-decoder
- It consists of two RNNs:
  - The **encoder** is an RNN responsible of processing the input element by element, storing in its internal state a compact and representative encoding of the information processed so far. The internal state of the encoder's neurons is called context vector or latent vector.
  - The **decoder** is another RNN that starts from the context vector of the last encoder state. In each step, it predicts an element of the output domain, using the internal state and the last element predicted. The process ends when a special element *EOS* (end of sentence) is emitted.

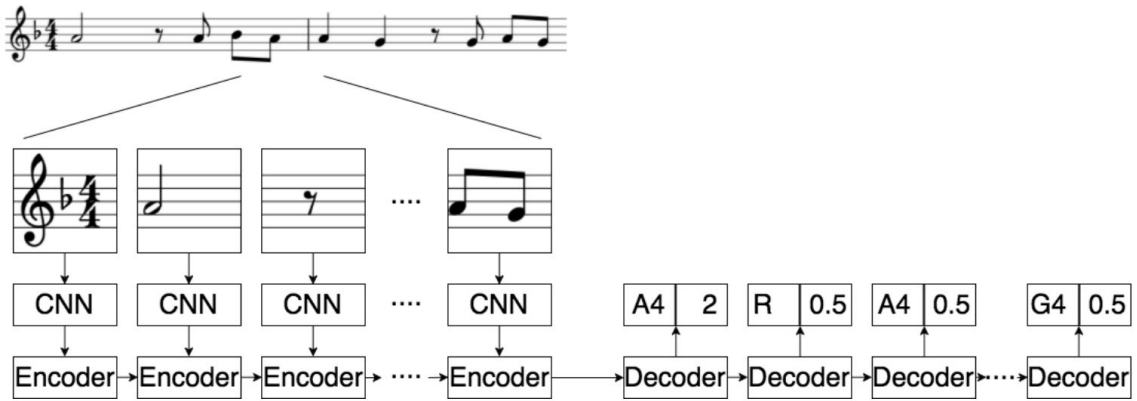
Encoder-Decoder Recurrent Neural Network

- It is used for sequence-to-sequence learning

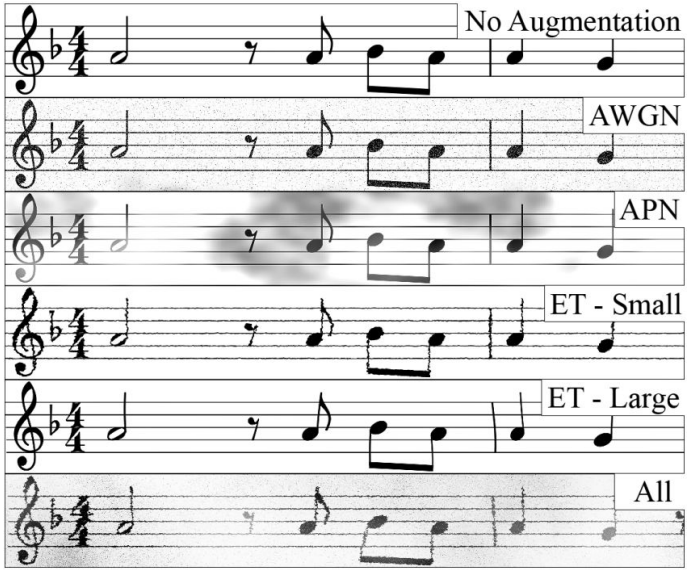


Encoder-Decoder CRNN

- Encoder-decoder approach + CNN
  - CNN-Encoder processes the input score image through a sliding window
  - Decoder outputs a musical symbol step by step
- Each output symbol consists of two components
  - Duration: codified respect to the beat
  - Pitch: absolute codification



Encoder-Decoder CRNN



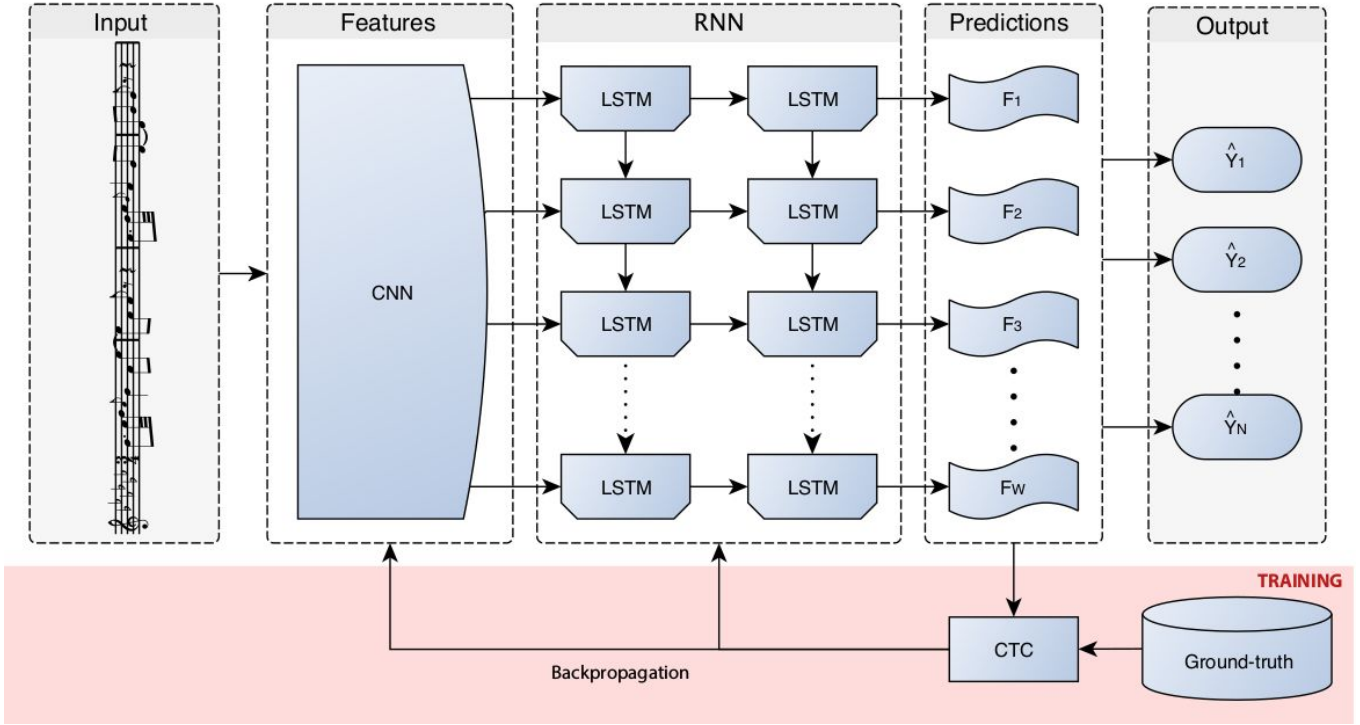
Training Augmentation	Pitch Accuracy	Duration Accuracy	Note Accuracy
AWGN	0.79	0.90	0.75
APN	0.81	0.89	0.76
ET - Small	0.78	0.89	0.74
ET - Large	0.78	0.94	0.75
All augmentations	0.79	0.92	0.77

CRNN + CTC

Approach based on the state of the art for text recognition:

- Convolutional Recurrent Neural Network (CRNN)
  - CNNs process the input image
  - RNNs takes care of the sequential nature of the task
  - Each frame is classified as one of the possible music-notation symbols
- Connectionist Temporal Classification (CTC)
  - Loss function that avoids the need of providing the location of the symbols during training
  - It is not optimal but approximated by means of an Expectation-Maximization approach

CRNN + CTC



Calvo-Zaragoza et al., 2017

CRNN + CTC

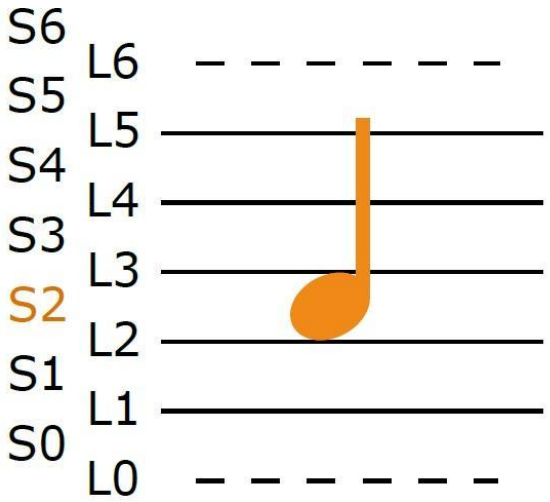
Internal representation

- How does the output look like?
  - It is often forgotten that OMR (fundamentally) belongs to the graphics recognition discipline
  - That is why, internally, the model prefers to learn graphic concepts instead of musical concepts.
  - For the computational learning process, it is better to use an internal representation, that we call agnostic, as opposed to a semantic representation.

CRNN + CTC

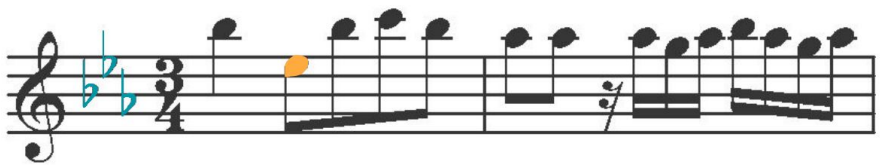
Internal representation

- Graphically, there are no pitches but positions within the staff lines



CRNN + CTC

Internal representation



- Semantic sequence
  - clef-G2 keySignature-EbM note-Bb5_quarter note-Eb5_eighth ...
- Agnostic sequence
  - clef.G-L2 flat-L3 flat-S4 flat-S2 note.black-S6 note.beamedRight-S4 ...

CRNN + CTC

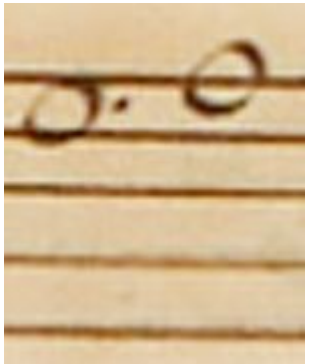
Internal representation

- Models that learn from agnostic representation are more robust and require fewer training data
- If we already have data available in semantic representation (MusicXML, MEI, Kern, ...) it should be easy to automate the conversion to agnostic representation
- With enough data, a deep model might also be capable of learning from any semantic representation

CRNN + CTC

Internal representation

- Additional advantages of agnostic representations

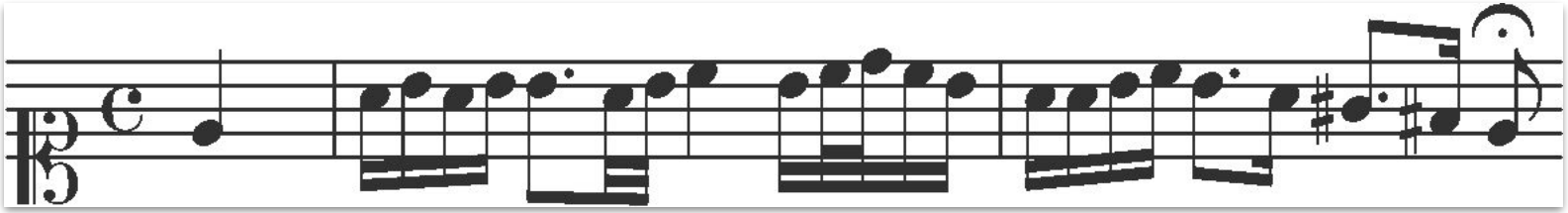


It allows annotation by non-experts

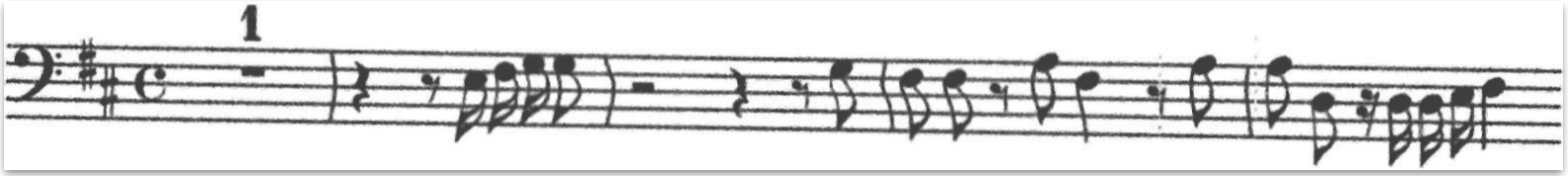
- What does this ‘dot’ mean?
  - *Punctus separationis*
  - *Punctus augmentationis*
- An agnostic representation does not care:
  - *Punctus*

CRNN + CTC

Experiments (Calvo-Zaragoza and Rizo, 2018):



Printed clean images (~99% accuracy at symbol level)

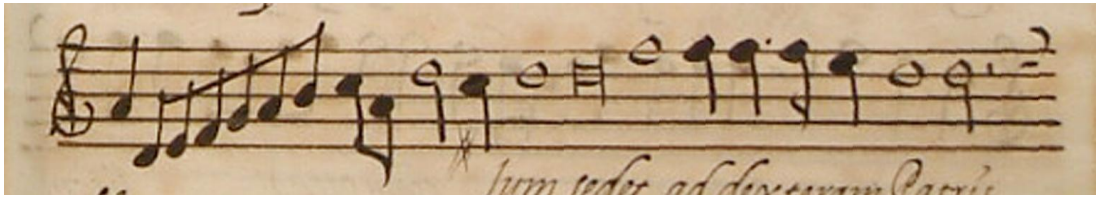


Printed camera-based (~98% accuracy at symbol level)

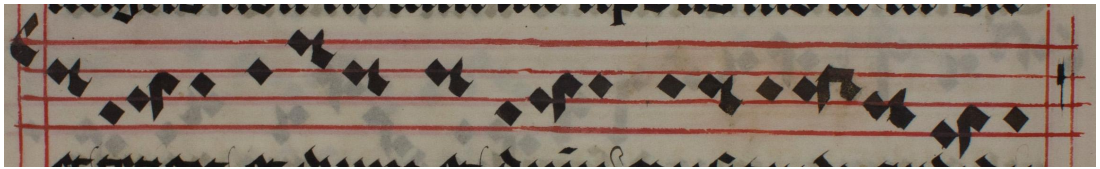
+ Data Augmentation!

CRNN + CTC

More experiments (unpublished):



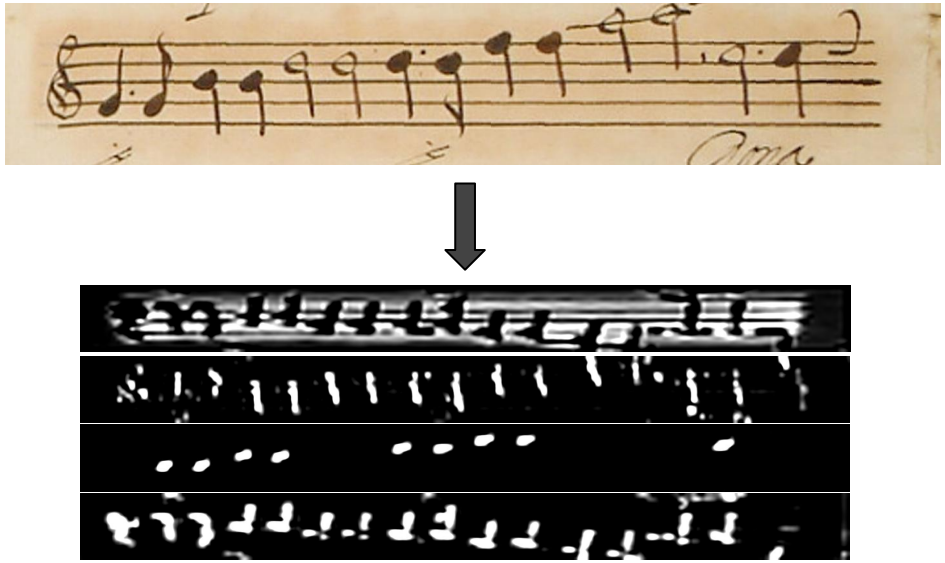
Handwritten mensural notation (~92% accuracy at symbol level)



Neumes (~99% accuracy at symbol level)

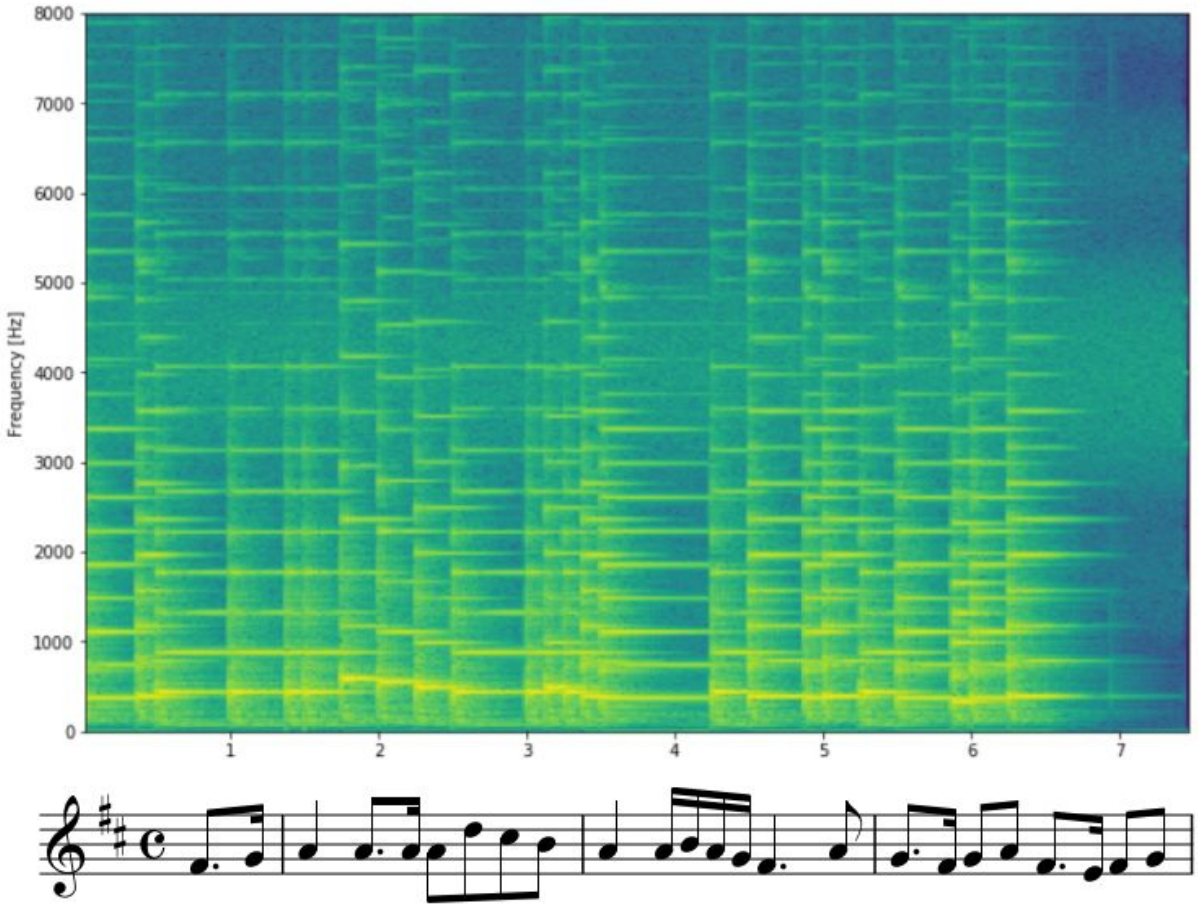
CRNN + CTC

What does the CNN do before the RNN?



NO OMF

- This CRN
  Automati
- The inpu



$ q c F♯4♪. G4♪ | A4♩ a4♪. A4♪ A4♪ D5♪ C♯5♪ B4♪ | A4♩ A4♪ B4♪ A4♪ G4♪ F♯4♩. A4♪ |
G4♪. F♯4♪ G4♪ A4♪ F♯4♪. E4♪ F♯4♪ G4♪

End-to-end OMR

Comparison among the end-to-end OMR proposals

- Fair comparison not possible: these works have experimented with their own datasets (different composition, different size, different ground-truth formats).
- An experimental comparison under the same conditions would be enlightening to know the pros and cons of each one.

End-to-end OMR

- The end-to-end approach is recent in the OMR field.
- It promises great benefits:
  - It works better than multi-stage approaches.
  - The system is more generalizable (training data is enough to adapt it to other domains)
  - Ground-truth data is less demanding (no geometrical information is required).
  - It naturally integrates contextual information that lead to the hypothesis that are more likely *a priori.*

End-to-end OMR

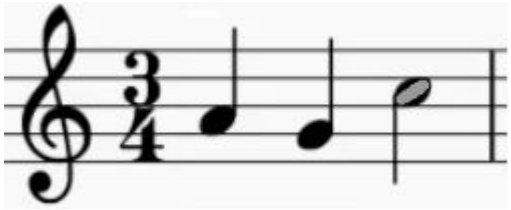
- The operation of these models is statistical
  - Given an input, it computes a probability for each possible hypothesis
  - Bayes' theorem is used to retrieve the most likely hypothesis:

$$\hat{s} = \arg\max_{s \in S} P(x|s)P(s)$$

x denotes the input
S is the set of all possible sequences
ŝ is the sequence predicted by the system

End-to-end OMR

- The most relevant aspect of the previous equation is to notice that the decision is based on two terms:
  - The probability of a sequence conditioned to the input
  - The prior probability of an output sequence
- Traditional systems only care about the probability conditioned to the input

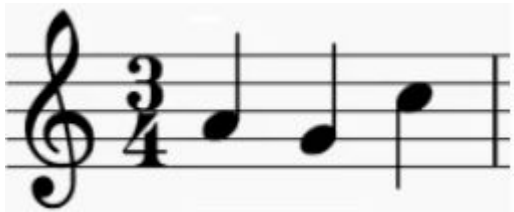


?

s_1

$P(x|s_1) = 60\ \%$

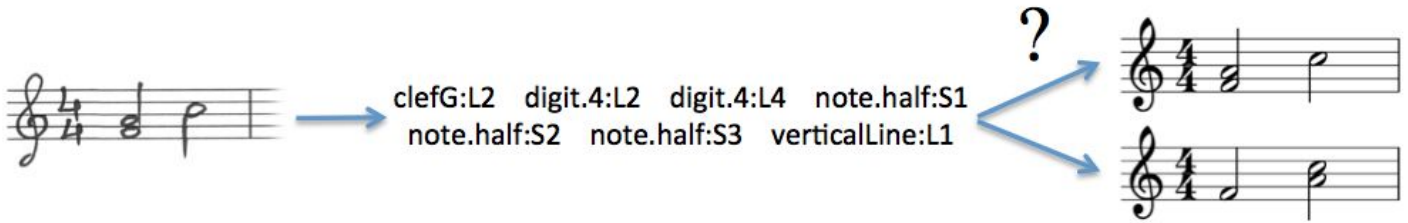
s_2



$P(x|s_2) = 40\ \%$

Recent advances (2019)

- The previous models provide a unidimensional sequence because of RNN design limitations
- Therefore, they only deal with monophonic scores
  - The output sequence can be parsed unambiguously
- If we increase the structural complexity by, for example, adding chords...



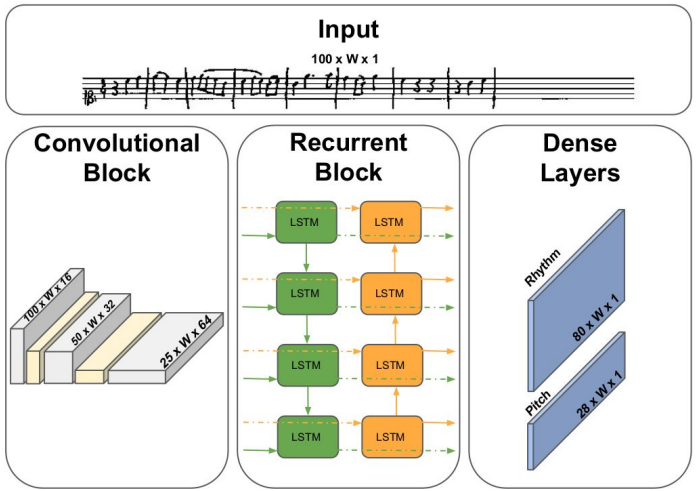
Recent advances (2019)

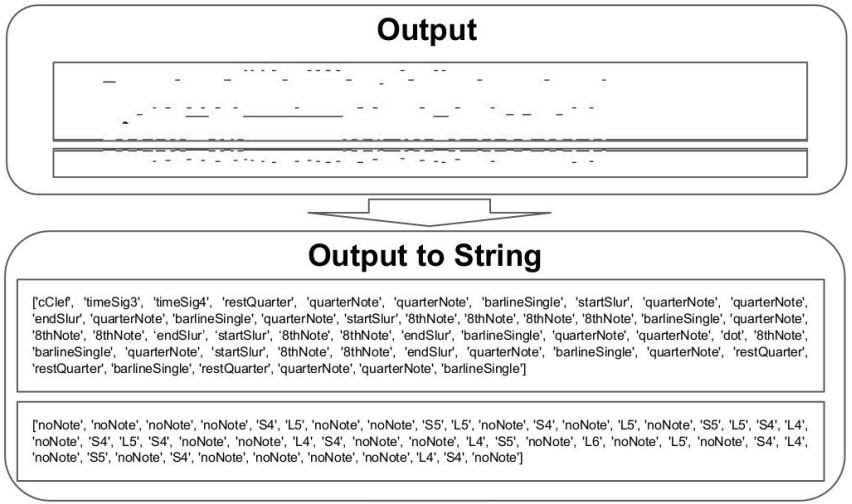
- Recently, two approaches have been proposed to allow for homophonic or single-staff polyphonic scores:
  - (Baro et al., 2019): Framewise multi-class CRNN
  - (Alfaro-Contreras et al., 2019): enriching the output sequence with ‘geometrical’ information

Framewise CRNN

CRNN approach with conventional framewise training mechanism:

- Advantage: it allows disentangling pitch and rhythm components
- Disadvantage: it requires a framewise ground-truth





Baro et al., 2018

Framewise CRNN

Limitations: when two “compatible” events occur at the same frame

Enriching the output sequence

- A **sequence of symbols** is not enough to unambiguously represent *complex* scores.
- Enrich the output sequence with symbols that denote geometrical information
- Four proposals:
  - **Remain-at-position character code**: symbols are assumed left to right, except when separated by a special element (*/*)
  - **Advance position character code**: symbols are assumed to be in the same vertical position except when separated by a special element (+)
  - **Parenthesized code**: when a vertical distribution appears in the score, the system outputs a *parenthesized* structure with two special elements (*vertical.start* ... *vertical.end*)
  - **Verbose code**: a combination of the two first ones

Enriching the output sequence

Remain-at-position character

clef.G:L2 accidental.flat:L3 digit.4:L2 / digit.3:L4 rest.eighth:L3 dot:S3 slur.start:S2 / note.quarter:S2 slur.end:S2 / note.sixteenth:S2 note.quarter:S2 verticalLine:L1 note.quarter:L1 / note.quarter:L2 / note.quarter:L3 note.beamedRight2:S2 note.beamedBoth2:L2 note.beamedBoth2:S1 note.beamedLeft2:L1 note.quarter:L1 verticalLine:L1 note.quarter:L1 / bracket.start-S6 note.quarter:S1 / digit.3-S6 note.quarter:L1 / bracket.end-S6 note.quarter:S1 verticalLine:L1

Enriching the output sequence

Advance position character

clef.G:L2 + accidental.flat:L3 + digit.4:L2 digit.3:L4 + rest.eighth:L3 + dot:S3 + slur.start:S2 note.quarter:S2 slur.end:S2 note.sixteenth:S2 + note.quarter:S2 + verticalLine:L1 + note.quarter:L1 note.quarter:L2 note.quarter:L3 + note.beamedRight2:S2 + note.beamedBoth2:L2 + note.beamedBoth2:S1 note.beamedLeft2:L1 + note.quarter:L1 + verticalLine:L1 + note.quarter:L1 bracket.start-S6 + note.quarter:S1 digit.3-S6 + note.quarter:L1 bracket.end-S6 + note.quarter:S1 + verticalLine:L1

Enriching the output sequence

Parenthesized

clef.G:L2 accidental.flat:L3 vertical.start digit.4:L2 digit.2:L4 vertical.end rest.eighth:L3 dot:S3 vertical.start slur.start:S2 note.quarter:S2 vertical.end vertical.start slur.end:S2 note.sixteenth:S2 vertical.end verticalLine:L1 vertical.start note.quarter:L1 note.quarter:L2 vertical.end note.beamedRight:S2 note.beamedLeft:L2 verticalLine:L1 vertical.start note.quarter:L1 bracket.start-S6 vertical.end vertical.start note.quarter:S1 digit.3-S6 vertical.end vertical.start note.quarter:L1 bracket.end-S6 vertical.end verticalLine:L1

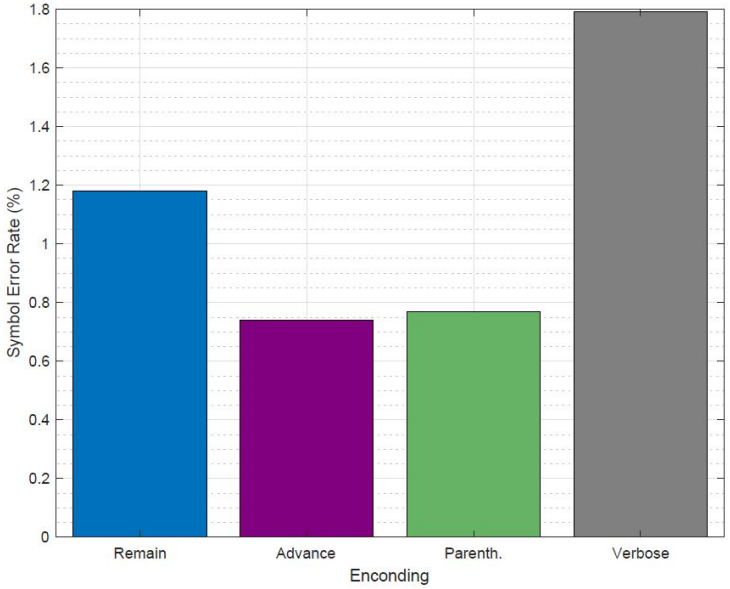
Enriching the output sequence

Verbose

clef.G:L2 + accidental.flat:L3 + digit.4:L2 / digit.3:L4 + rest.eighth:L3 + dot:S3 + slur.start:S2 / note.quarter:S2 + slur.end:S2 / note.sixteenth:S2 + note.quarter:S2 + verticalLine:L1 + note.quarter:L1 / note.quarter:L2 / note.quarter:L3 + note.beamedRight2:S2 +note.beamedBoth2:L2 + note.beamedBoth2:S1 + note.beamedLeft2:L1 + note.quarter:L1 + verticalLine:L1 + note.quarter:L1 / bracket.start-S6 + note.quarter:S1 / digit.3-S6 + note.quarter:L1 / bracket.end-S6 + note.quarter:S1 + verticalLine:L1

Enriching the output sequence

Piece of cake for the CRNN...



- 8,000 training staves
- 2,000 test sequence

Enriching the output sequence

Example of transcription for Incipit RISM ID no. 000136642-1_1_1

clef.G:L2 + accidental.sharp:L5 + accidental.sharp:S3 + accidental.sharp:S5 + digit.8:L2 digit.6:L4 + note.beamedRight2:S2 + note.beamedRight1:S3 + note.beamedLeft1:L4 + note.quarter:S4 + note.eighth:S3 + verticalLine:L1 + note.quarter:L4 note.quarter:L5 + dot:S4 [dot:S5] + [note.quarter:S3] note.quarter:S4 + [dot:S3] dot:S4 + verticalLine:L1 + rest.eighth:L3 + note.beamedRight1:L4 + note.beamedLeft1:S3 + note.quarter:L3 + note.eighth:S2 + verticalLine:L1 + note.beamedRight1:L2 + note.beamedRight1:S2 + note.beamedLeft1:L3 + note.quarter:L1 note.quarter:L2 note.quarter:L3 + dot:S1 dot:S2 dot:S3

Next steps

Next steps

- Recent advances have moved OMR research from sub-problems to complete systems
- Two lines of research:
  - Music-Object Detection + Notation Assembly
  - End-to-end systems
- Still a lot of work to do:
  - Domain adaptation / Transferability across input domains
  - Standardize *structured encoding* application
    - Evaluation
    - Internal representation
  - Syntactic and semantic modelling
  - Page-level end-to-end OMR

Session III
OMR system with Deep Learning

Materials

Google Colaboratory - Google Drive Notebooks

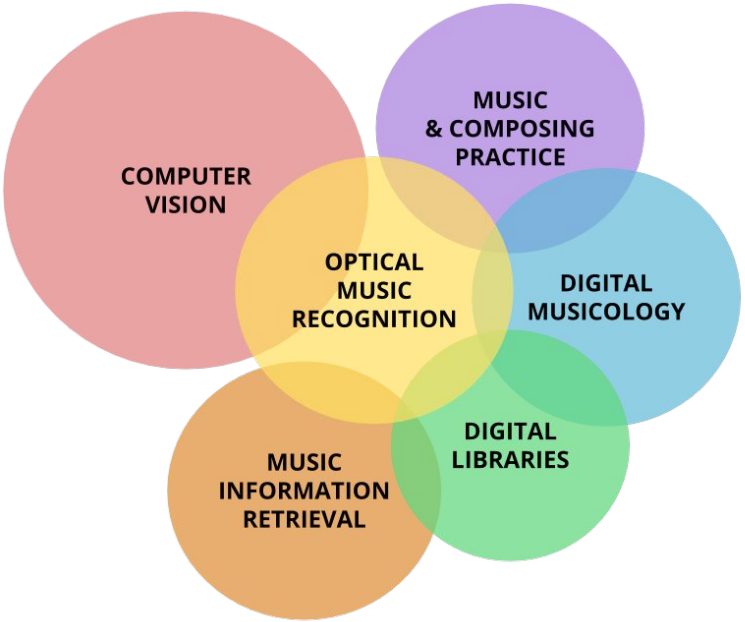
- Step-by-step (simple) Optical Music Recognition system

Session IV
Closing remarks

Community

Venues

- OMR involves several disciplines
  - Stakeholders: music information retrieval and digital libraries
  - Methodology: machine learning, pattern recognition and document analysis

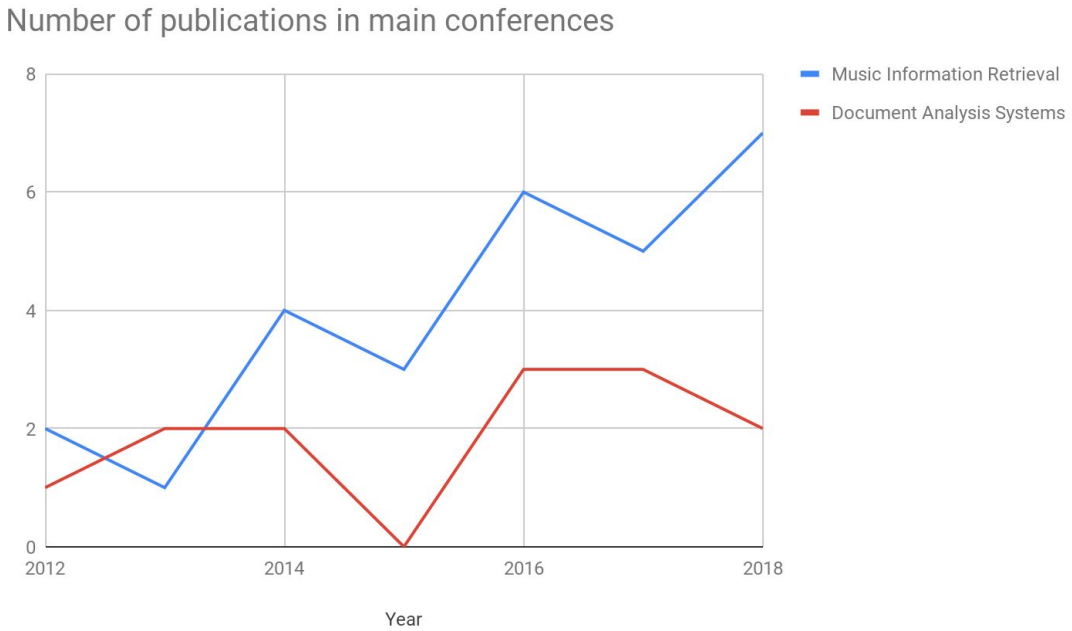


Venues

- OMR publications fall in a number of different venues
  - Music Information Retrieval and Music Digital Libraries
    - International Society for Music Information Retrieval conference (ISMIR)
    - Digital Libraries for Musicology (DLfM)
    - Journal of New Music Research (JNMR)
  - Document Analysis and Computer Vision
    - International Conference on Document Analysis and Recognition (ICDAR)
    - Document Analysis Systems (DAS)
    - International Journal on Document Analysis and Recognition (IJDAR)
    - International Conference on Pattern Recognition (ICPR)
    - Pattern Recognition Journal

Venues

- There has been a lack of stable, focused community around OMR
- Publications are rather scattered (and few)



Venues

- The International Workshop on Reading Music Systems (WoRMS) took place for the first time, as a satellite event of ISMIR 2018
  - Around 30 attendees
  - 12 papers covering most of the aspects related to OMR
    - Community
    - Applications and Interactive Systems
    - Technical solutions
    - User perspectives
- Follow-up
  - Next edition will most likely be co-located with ISMIR 2019

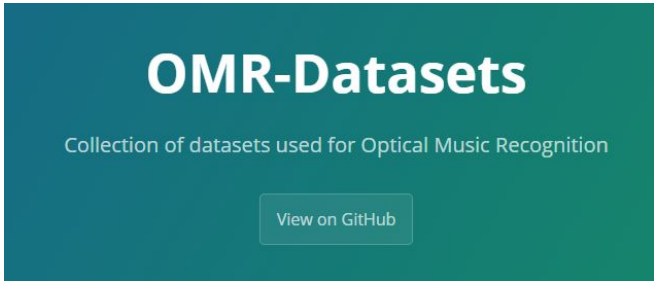
The OMR Research community

- The website of the OMR Research community was recently launched:
  - https://omr-research.net/
  - And its Twitter account: https://twitter.com/OMR_Research

Resources

Datasets

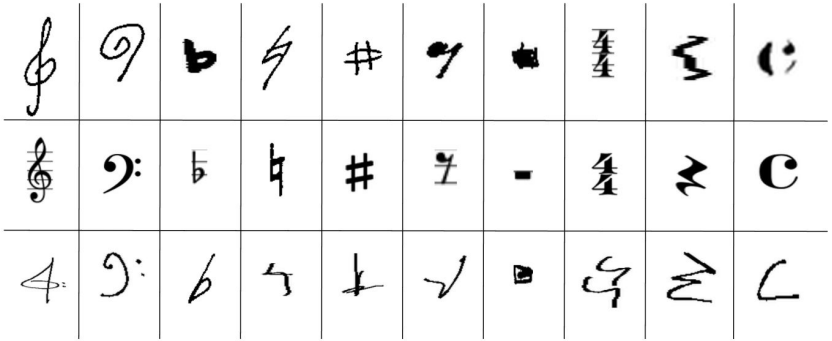
- >20 datasets listed on OMR Datasets project website
- Mostly created by academia and freely available for research



http://apacha.github.io/OMR-Datasets

Datasets for Symbol Classification

- **H**andwritten **O**nline **Mu**sic **S**ymbols (HOMUS)
- MUSCIMA++
- Capitan collection
- Rebelo Dataset
- Fornés Dataset
- Audiveris OMR Dataset
- DeepScores

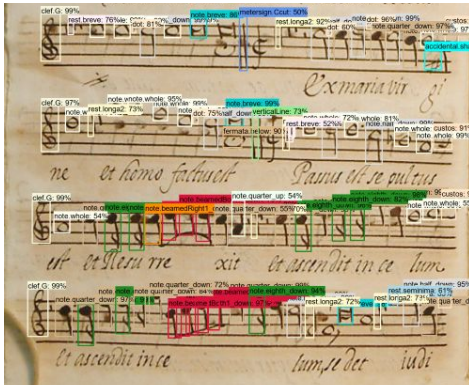


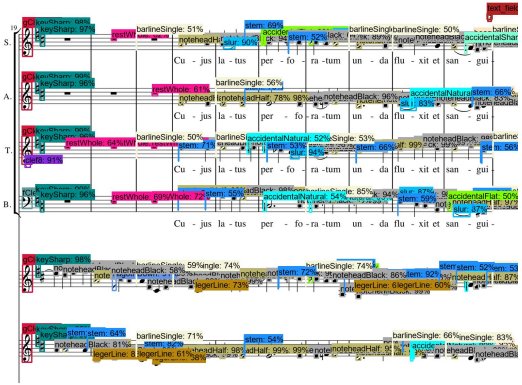
And even more...

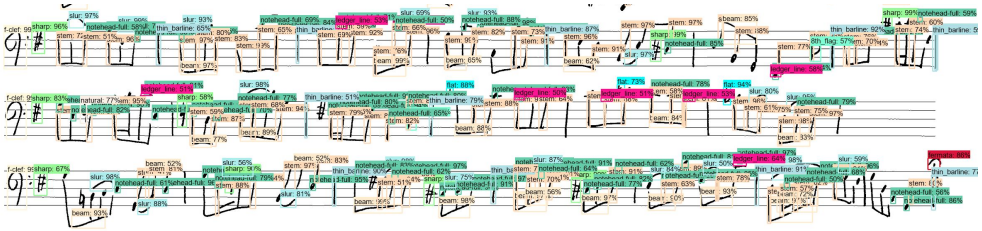
Many of them are unified in the **Universal Music Symbol Collection**

Datasets for Music Object Detection

- MUSCIMA++
  - Handwritten scores
  - Modern notation
  - 140 annotated pages
- Capitan collection
  - Handwritten scores
  - Mensural notation
  - 46 annotated pages
- DeepScores
  - Printed scores
  - Modern notation
  - 1700 annotated pages

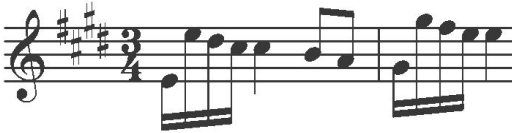


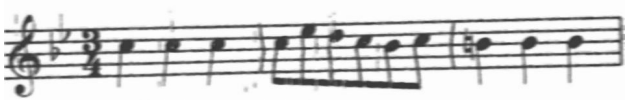


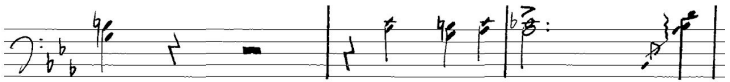


Datasets for End-to-End Recognition

- PrIMuS & Camera-PrIMuS
- Baro’s (Synthetic Handwritten)



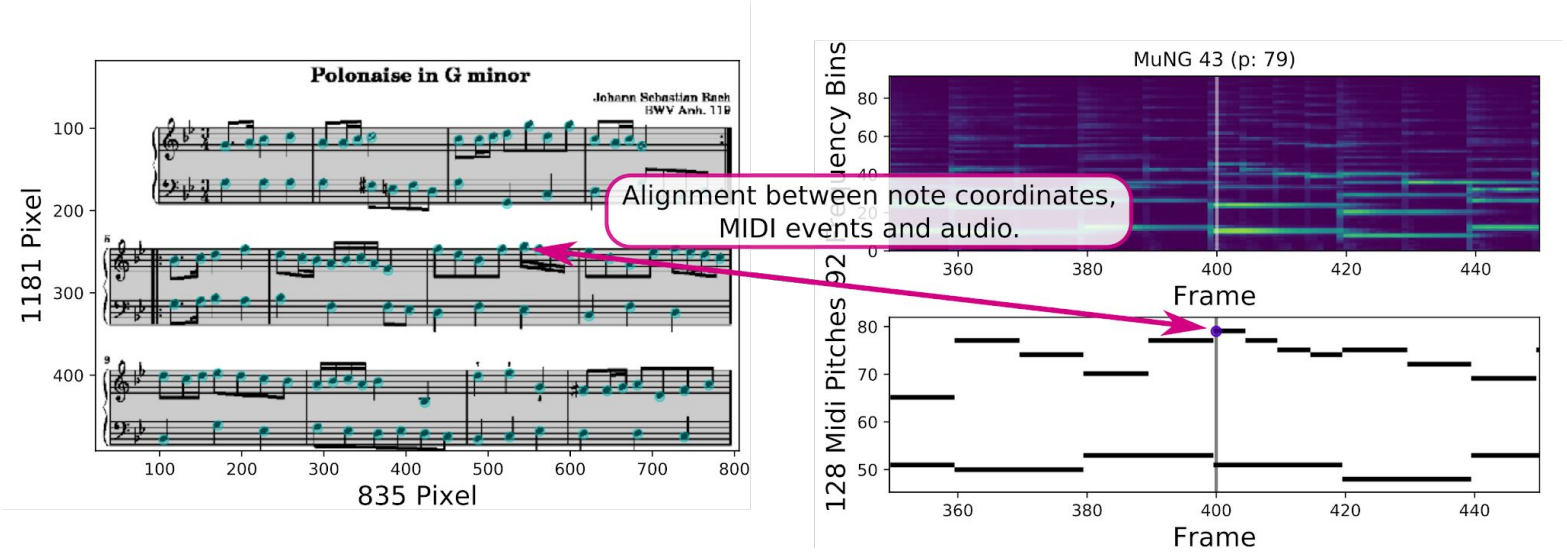




```
--------------------
|w-15_p-10_s-6.png |
--------------------
fClef   noNote
accidentalFlat  L2
accidentalFlat  S3
accidentalFlat  S1
accidentalNatural   S5
startChor   noNote
quarterNote S5
quarterNote S4
endChor noNote
...
```

Multi-Modal datasets

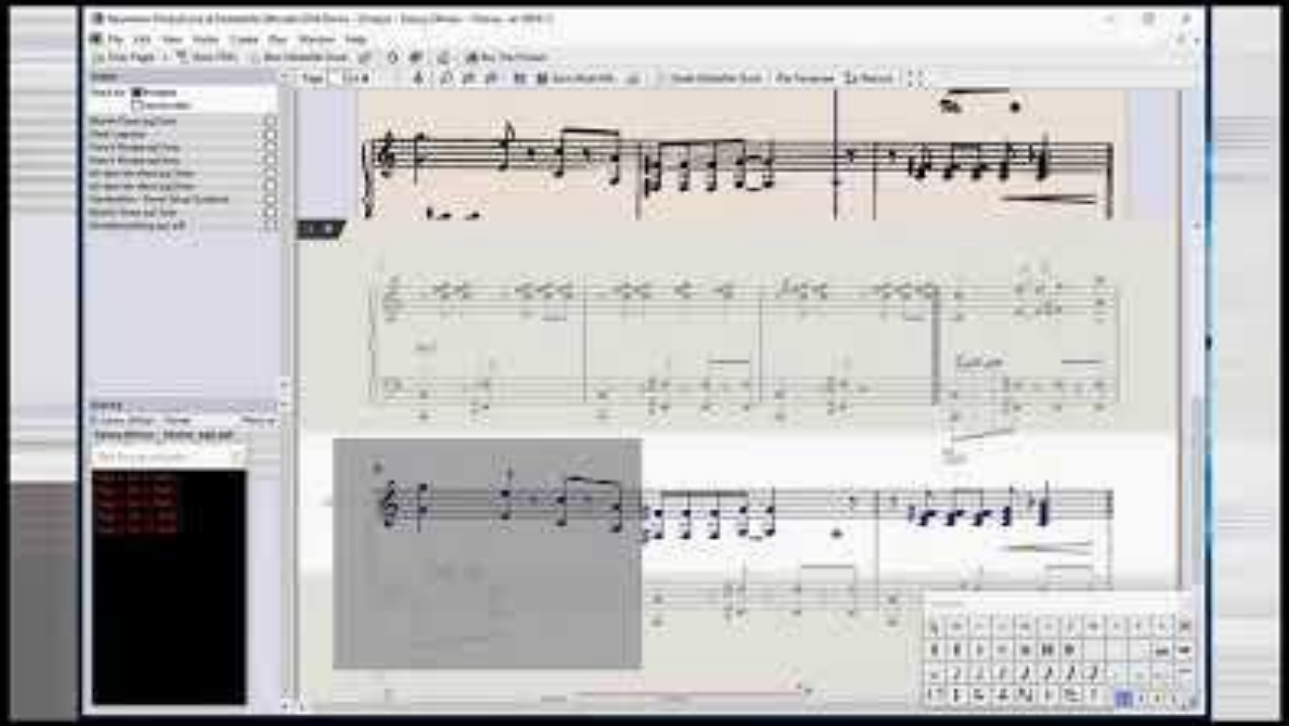
- Multimodal Sheet Music Dataset (Dorfer et al., 2018)
  - https://github.com/CPJKU/msmd
  - 300,000+ aligned notehead-MIDI event files
  - Used for live score following and cross-modal retrieval



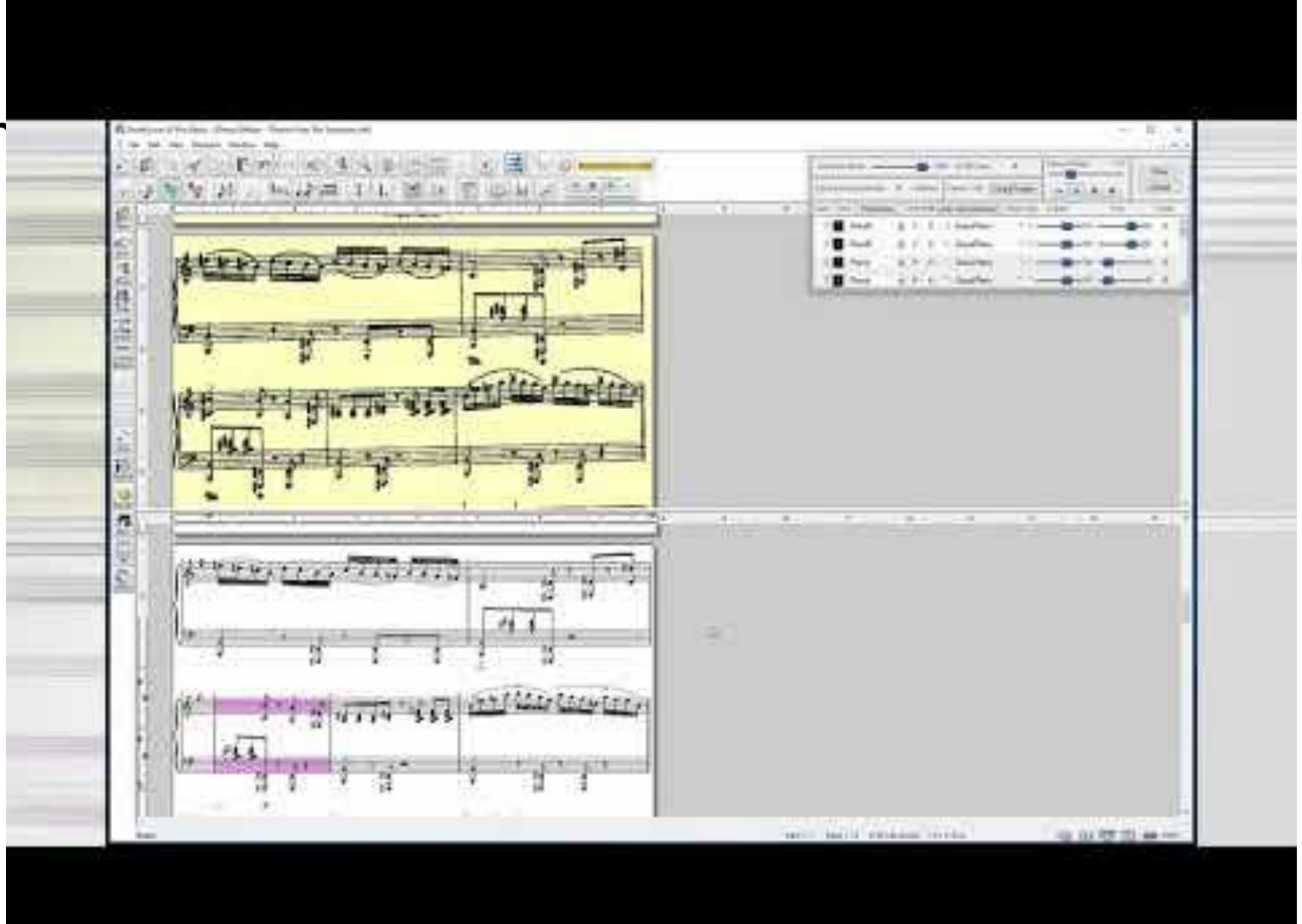
Commercial applications

- Old Guard:
  - SmartScore, Neuratron PhotoScore, SharpEye (☥ RIP)
- New Blood:
  - NotateMe, StaffPad - pen-based recognition
- Mobile apps:
  - PlayScore, iSeeNotes, KompApp
- Old Guard is decent for born-digital and nicely scanned PDFs
- Not much information on New Blood performance

C



Cor



Collection of Repositories for OMR

- https://github.com/tuggeluk/DeepWatershedDetection
- https://github.com/tuggeluk/DeepScoresExamples
- https://github.com/apacha/MusicObjectDetector-TF
- https://github.com/apacha/MusicSymbolClassifier
- https://github.com/apacha/Mensural-Detector
- https://github.com/Audiveris/audiveris
- http://ddmal.github.io/Rodan/
- https://github.com/tensorflow/moonlight
- https://github.com/calvozaragoza/tf-deep-omr
- https://github.com/hajicj/muscima
- https://github.com/greenjava/OpenOMR
- https://github.com/DDMAL/aruspix

OMR is already useful: a case of study

Is OMR useful in its current state? A case of study

- A case of study: use of (non-perfect) OMR results to demonstrate that systems can be useful in their current state
- Three challenges over written sources:
  - Search for copies
  - Melodic similarity
  - Digital musicology
- Using scientific OMR contributions (end-to-end: CRNN+CTC)

Is OMR useful in its current state? A case of study

- Experimental data set
  - 10.000 incipits of the RISM database of modern notation
  - Random duplicates by synthetically altering the conditions of the image
  - Two twin MIDI-based sets to compare:
    - Real MIDIs
    - MIDIs generated by OMR

Is OMR useful in its current state? A case of study

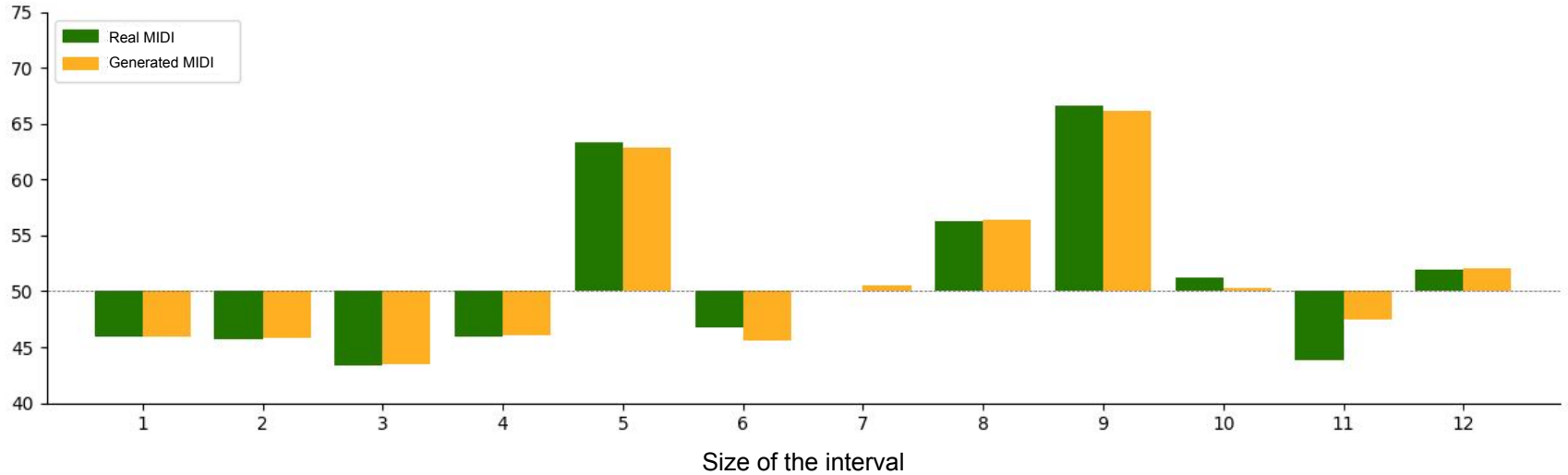
- **First challenge.** Search for copies: given an image of an incipit, find if there are one (or several) copies of it.
  - From the image of the incipit, a MIDI file is generated with OMR
  - We seek for copies within the set that contains the real MIDIs.
    - First result: 100% success.
    - All results: 99.8% success.
  - We seek for copies within the set containing the MIDIs generated by OMR.
    - First result: 100% success.
    - All results: 99.8% success.

Is OMR useful in its current state? A case of study

- **Second challenge.** Melodic similarity: compute the similarity of a given MIDI query against the set of generated MIDIs and the set of real MIDIs.
  - We use a widely known metric of the literature (Urbano, 2013)
  - We obtain two set of metrics and we want to compute the relationship between them
  - Results:
    - Spearman coeff.: the relationship is monotonous
      - Value of 94 %
    - Pearson coeff.: the relationship is linearly correlated
      - Value of 97 %

Is OMR useful in its current state? A case of study

- **Third challenge.** Digital musicology: characterize a music collection based on the frequency in the size of the intervals (Vos and Troost, 1989)
  - Would we draw the same conclusions from OMR-based symbolic music?



Is OMR useful in its current state? A case of study

- In summary, this case of study show that:
  - OMR might be functional enough to perform certain case studies.
  - In most situations, the OMR does not produce exactly the same results, but probably similar enough to draw equal conclusions
  - Therefore, OMR **already** opens the possibility of interesting "*distant reading"* experiments without the need to enter the data manually

Conclusion

Closing

- Machine Learning approaches, supported by recent advances in Deep Learning, had an impact on OMR research
- OMR is becoming useful for more and more practical purposes (Hajic et al., 2018b; Rizo et al., 2018)
  - Generic machine learning makes OMR methods transfer easier to new use-cases
- The community is growing
  - Public datasets, code repositories, … and WoRMS!
- Good time to dive into OMR

Acknowledgements

- I would like to thank my colleagues **Jan Hajic Jr.** and **Alexander Pacha**, as well as **Prof. Ichiro Fujinaga**, who contributed to the materials presented in this session.
- And of course, thank **YOU** for attending this course!

Optical Music Recognition

Jorge Calvo Zaragoza

Sound and Music Computing Summer School
Málaga (Spain) May 26, 2019