INTRODUCTION AND RELATED WORKS

TRADITIONAL STEPS IN OMR

1. **Image preprocessing**: tasks such as binarization, distortion correction, or stave separation.
2. **Music symbol detection**: bounding box detection and classification.
3. **Notation assembly**: where the independent components are related to each other to reconstruct the musical notation.
4. **Encoding**: in which the recognized notation is exported to a specific language that can be stored and further processed by computational means.

Related works

- **Only one existing work** focused on the relationship retrieval: Pacha et al [1].
- Based on a Convolutional Neural Network (CNN).
- **Tremendously inefficient**: requires the independent construction and classification of an image for each pair of nodes.

PROBLEM FORMULATION

The notation assembly stage can be seen as a relationship predictor of a graph where the nodes are the music symbols and the edges are the relations between the nodes.

RELATIONSHIP PREDICTION PROPOSALS

- **Node**: 20-dimensional feature vector:
  - Bounding box: top-left and bottom-right normalized values
  - Class information: 16-dimensional learnable embedding layer

- **MLP_{AsymK}**: A three-layered fully-connected network comprising two hidden layers with 64 and 512, respectively, with Rectifier Linear Unit (ReLU) activations and a single output unit to compute the score of the binary classification.
- **MLP_{2L}**: A two-layered fully-connected network comprising a 32-unit hidden layer and ReLu activation and a single unit as output.
- **Asymmetric kernels**: are implemented as two different 4-layered MLP comprising 512, 1024, 512, and 256 units, respectively, with ReLU activation. The idea is to generate two 256-dimensional embeddings—two points in different Hilbert spaces—to then compute the similarity through the dot product.

EFFICIENCY

<table>
<thead>
<tr>
<th>Execution time (ms)</th>
<th>AsymK</th>
<th>MLP_{32}</th>
<th>MLP_{64,512}</th>
<th>CNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;0.5</td>
<td>55</td>
<td>176</td>
<td>&gt;1.5·10^{6}</td>
<td></td>
</tr>
</tbody>
</table>

Efficiency results in terms of the per-page absolute execution time (in milliseconds) on the MUSCIMA++ corpus for the different notation assembly methods assessed. Each value corresponds to the average execution time obtained with 10 different iterations over all test samples.

RESULTS AND DISCUSSION

EFFICACY

ACKNOWLEDGEMENTS: Work produced with the support of a 2021 Leonardo Grant for Researchers and Cultural Creators, BBVA Foundation. The foundation takes no responsibility for the opinions, statements, and contents of this project, which are entirely the responsibility of its authors.