Pixel.js: Web-based Pixel Classification Correction Platform for Ground Truth Creation

Zeyad Saleh, Ké Zhang, Jorge Calvo-Zaragoza, Gabriel Vigliensoni, and Ichiro Fujinaga
Distributed Digital Music Archives & Libraries Lab (DDMAL)
McGill University
Montréal, Canada
Context: The SIMSSA Project

SIMSSA: Single Interface for Music Score Searching and Analysis

Social Sciences and Humanities Research Council of Canada

Teach computers to recognize the musical symbols in digital images of musical scores
Context: The SIMSSA Project

Digital Manuscript

Image Segmentation

Semantic Labelling

Interpretation

Correction

Symbolic Representation

Machine Learning

For more information:
https://simssa.ca/
Motivation

Separate elements of document images such as:
Motivation

Separate elements of document images such as:

- Text
Motivation

Separate elements of document images such as:

- Text
- Glyphs
Motivation

Separate elements of document images such as:

- Text
- Glyphs
- Staff Lines
Motivation

Separate elements of document images such as:

- Text
- Glyphs
- Staff Lines

Process the different elements of the document image separately
Motivation: Heuristic Algorithms

Original image

Calvo-Zaragoza et al.
2016
Motivation: Heuristic Algorithms

Original image

Heuristic Binarization (Sauvola)

Calvo-Zaragoza et al. 2016
Motivation: Heuristic Algorithms

Use previous results of automatic, pixel-level image segmentation algorithms and correct its misclassified pixels to produce ground truth

(Instead of creating it from scratch)
Motivation

Ground truth in this context refers to creating multiple layers of information
An online platform for correcting the results of image classification algorithms at pixel-level
Nomen Domini secutus est de longinquo et claritas eius rosiorem terrarum. Et nos.
Representation:
- Layers
- Colour-coding
Uploading images to layers

- Automatic colour conversion
- Transparent background required
Pixel.js: Tools & Functionalities

Zooming & Navigation

Notice misclassified pixels:
Quickly move regions of pixels from a layer to another
Changing layer’s opacity
Drawing & Erasing

Changing brush size
Exporting

- Can re-import to continue work
- 3 different formats
Why Pixel.js?

- Music documents require a higher level of ground truth accuracy
- Opens up ground truthing to a larger audience
- Keyboard shortcuts built specifically for ground truth creation
## Existing Tools: Comparison

<table>
<thead>
<tr>
<th></th>
<th>Web-based</th>
<th>Open Source</th>
<th>Free to use</th>
<th>Freeform Labelling</th>
<th>Batch Labelling</th>
<th>Handles Large Images</th>
<th>Pixel-level Classification</th>
<th>Independent from preprocessed input</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel.js</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>PixLabeler</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Divadia</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Web-GT</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Photoshop / Pixelmator</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Picozu</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Diva.js

A web-based document viewer optimized for high-resolution image collections

- Supports IIIF
- 30 megapixels, ~180 MB per image
- 180MB x 479 images ≈ 86 GB
Classifying the different elements of the same music manuscript page using both Pixelmator and our tool Pixel.js
Positive feedback:

- Time efficiency: A reduction of 40% in production time
  - From 30 with Pixelmator to 18 hours/page with Pixel.js
Future Work

- Collaborative platform
- More targeted tools
Check out our Github repository

https://github.com/DDMAL/Pixel.js
Thank you!

https://github.com/DDMAL/Pixel.js