

Oct 10, 2025

CALLAF — Creation, Digitization, Vectorization, Indexing, and CGAN Validation

DOI

<https://dx.doi.org/10.17504/protocols.io.n2bvjeqbngk5/v1>

Siraj Allaf¹

¹King Abdulaziz University



SIRAJ ALLAF

King Abdul Aziz University

Create & collaborate more with a free account

Edit and publish protocols, collaborate in communities, share insights through comments, and track progress with run records.

Create free account

OPEN  ACCESS



DOI: <https://dx.doi.org/10.17504/protocols.io.n2bvjeqbngk5/v1>

Protocol Citation: Siraj Allaf 2025. CALLAF — Creation, Digitization, Vectorization, Indexing, and CGAN Validation. **protocols.io** <https://dx.doi.org/10.17504/protocols.io.n2bvjeqbngk5/v1>

License: This is an open access protocol distributed under the terms of the **[Creative Commons Attribution License](#)**, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

We use this protocol and it's working



Created: October 05, 2025

Last Modified: October 10, 2025

Protocol Integer ID: 229039

Keywords: authentic arabic calligraphy artworks for inclusion, authentic arabic calligraphy artwork, callaf dataset, digitization, cgan validation

Abstract

To describe the complete, reproducible workflow used to digitize, preprocess, and vectorize authentic Arabic calligraphy artworks for inclusion in the CALLAF dataset.

Materials

Calligraphic material

- Artists: 13 professional calligraphers (styles: Dewani, Naskh, Reqaa)
- Tools: traditional Arabic calligraphy pens (reed/qalam), traditional ink
- Paper: high-quality glossy paper (A4)

Hardware

- Scanner: high-resolution flatbed, 300 DPI
- Workstation: MacBook Pro (M4 Pro)

Software

- Vectorization: Adobe Illustrator, Vector Magic, Potrace, scikit-image
- ML / CV: PyTorch, OpenCV
- SVG tooling: svgpathtools



Procedure

1 **Select and write items**

Isolated characters and word forms representing the majority of Arabic calligraphic styles are selected and formatted on A4 sheets. Each calligrapher is tasked with writing these isolated characters and words in Dewani, Naskh, and Reqaa types, ensuring precise composition and consistent baselines.

2 **Scan & clean**

Scan at 300 DPI, crop borders, fix orientation, clean artifacts, convert to transparent grayscale PNG.

Naming:

```
type_category_resolution_index
```

(e.g.,

```
dew_char_1024_00001.png
```

3 **Normalize & multi-resolution**

Normalize canvases to 1024×1024 px. Downscale to 256, 128, 64, 32 px.

4 **Vectorization**

From each 1024 px master, generate four SVGs using Adobe Illustrator, Vector Magic, Potrace, scikit-image. Normalize viewBox/units so SVGs overlay rasters exactly.

5 **Directory layout**

Organize as:

```
raster/all/  
  /                # 32 | 64 | 128 | 256 | 1024  
    *.png          # named:  
type_category_resolution_index.png
```

,



```

raster/split/
  /          # 32 | 64 | 128 | 256 | 1024
    /          # dew | nas | req
      /          # char | word
        /          # e.g., baa, meem, ain_alef, rahma
          *.png          # named:
type_category_resolution_index.png

```

```

vector/all/1024/
  /          # illustrator | vector magic | potrace | skimage
    *.svg          # named:
type_category_resolution_index.svg

```

, and

```
csv/index_*
```

files.

6 **CSV indices**

Create CSVs listing filename, type, category, resolution, content, modality (and tool for vectors).

Files:

```
index_raster_all.csv
```

```
index_raster_split.csv
```

```
index_vector_all.csv
```

7 **Manual Quality Control**

Visually inspect random batches across all types and modalities. Verify labels, PNG-SVG alignment, and absence of artifacts. Cross-check CSV coverage and paths

8 **CGAN validation**



Train Conditional WGAN-GP (with spectral norm) on 64×64 grayscale, conditioned on type.

Report: SSIM_mean = 0.616, LPIPS_mean = 0.123, LPIPS_p95 = 0.282

Acknowledgements

Repository & License

All data and scripts are publicly available under CC0 1.0 Universal at CALLAF Dataset Repository (DOI: TBD).