Visão Computacional para Preservação Digital: Rumo à Automação da Catalogação de Tipos Móveis do Museu Paulista

Matheus Hencklein Ponte *m247277@dac.unicamp.br*
Solange Ferraz de Lima *sflima@usp.br*
Paula Dornhofer Paro Costa *paulad@unicamp.br*

Departamento de Engenharia de Computação e Automação (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)

Resumo

A curadoria, a classificação e, por fim, a catalogação de coleções históricas, são essenciais para a preservaão do patrimônio cultural, mas ainda costumam ser feitas manualmente, de forma lenta e suscetível a erros. Este trabalho é uma das iniciativas da parceria entre o DCA/FEEC/Unicamp e o Museu Paulista/USP que visa o desenvolvimento de ferramentas de inteligência artificial para apoiar o trabalho de museólogos e historiadores. Em particular, o trabalho toma como objeto de experimentação o projeto de catalogação da coleção tipográfica de Tércio Ferdinando Gaudêncio e explora modelos de visão computacional para identificar padrões visuais e organizar os tipos móveis e florões em seis famílias tipográficas: escritural, grotesca, serifada, fantasia, monograma e toscana. Mais especificamente, o presente artigo relata a construção de um banco de dados rotulado a partir de fotografias de amostras impressas, capturadas em condições variadas. As imagens passam por pré-processamento e segmentação, e os recortes resultantes são avaliados por voluntários. Os próximos passos incluem validar a segmentação, rotular os caracteres e consolidar o banco de dados que servirá ao treinamento de modelos de classificação.

Palavras-Chave — Tipografia, Identificação de Fontes, Famílias Tipográficas, Aprendizado de Máquina, Visão Computacional

1. Introdução

Em 2015, o Museu Paulista/USP recebeu a coleção tipográfica de Tércio Ferdinando Gaudêncio, composta por prensas, ferramentas de douração e encadernação, catálogos e cerca de 196 gavetas com material branco, tipos metálicos, fontes e ornamentos, que constituem o objeto deste projeto. A coleção reúne um conjunto raro de tipos móveis que preserva parte importante da história gráfica paulistana. Mas catalogar esse material é um trabalho lento e complexo: são centenas de gavetas, milhares de peças, muitas delas desgastadas, misturadas ou sem documentação (Figura 1). O processo depende de inspeção manual cuidadosa, o que dificulta a organização e o acesso ao acervo [1]. Nesse contexto, técnicas de visão computacional e IA podem apoiar o trabalho museológico, automatizando etapas repetitivas e acelerando a identificação dos caracteres. Isso não só facilita a preservação digital do acervo, mas também amplia as possibilidades de pesquisa e entendimento sobre a memória tipográfica brasileira.

O presente trabalho iniciou-se com uma visita técnica ao depósito dos acervos do Museu Paulista para o entendimento do processo de catalogação pela equipe de museólogos e historiadores do museu. Dessa visita, obteve-se o acesso a uma base de imagens fotográficas de amostras de "impressão" desses tipos em folhas de papel, isto é, *specimens* tipográficos produzidos pelos museólogos para registrar, lado a lado, os caracteres das diferentes famílias identificadas no acervo. Tais imagens fotográficas caracterizam a base de dados brutos (*raw data*) deste trabalho e são caracterizadas como imagens em escala de cinza de amostras de tipos móveis, cada uma correspondente a uma gaveta tipográfica. As amostras, impressas em preto sobre papel branco, em geral, incluem os seguintes caracteres: (1) letras maiúsculas: A, B, C, Q, R, S; (2) letras minúsculas: a, s, g; (3) números: 1, 2, 3; (4) caracter & ("E comercial").

Neste trabalho apresenta-se o processo de construção de uma base de dados de tipos rotulada, etapa fundamental para as fases seguintes do projeto, que envolvem a construção de modelos classificadores de tipos a partir de imagens.

2. Métodos

O *pipeline* de construção do banco de dados até o seu estado atual é apresentado na Figura 2.

Figura 1: Detalhe de uma gaveta com tipos móveis. O processo de catalogação envolve a "impressão" desses tipos em folhas de papel para posterior categorização e catalogação. Atualmente, esse trabalho manualmente e envolve o trabalho minucioso de inspeção visual.

2.1. Coleta e Organização dos Dados Brutos

Inicialmente, cada imagem foi manualmente extraída e nomeada no formato `cod_XXX.YYY`, em que `XXX` é um código numérico de três dígitos e `YYY` o formato da imagem. Amostras com múltiplos estilos tipográficos foram descartadas, resultando em 149 imagens válidas. As imagens restantes foram organizadas em pastas conforme o estilo já definido na planilha: escritural (50 imagens), fantasia (5), grotesco (60) e serifado (34), armazenadas no diretório `raw`.

2.2. Aplicação de Transformações nas Imagens

As amostras foram coletadas em condições variadas e apresentam ruídos devido a manchas de tinta e falhas de impressão, semelhantes a ruído impulsivo [2, p. 207], baixo contraste entre caractere e fundo e símbolos parcialmente desconexos. Por isso, o pré-processamento tornou-se necessário, e suas etapas são descritas a seguir. As imagens intermediárias foram armazenadas em `interim`.

2.2.1. Remoção de Ruído e Binarização

Para reduzir o ruído impulsivo, aplicou-se um filtro de mediana. Seja $g(x, y)$ a imagem original contaminada por ruído, e seja $S_{x,y}$ a vizinhança quadrada centrada no pixel (x, y), composta pelos pares de coordenadas (s, t) pertencentes à janela de 15×15 pixels. A imagem filtrada $\hat{f}(x, y)$ é dada por:

$$\hat{f}(x, y) = \text{mediana}\{\, g(s, t) \mid (s, t) \in S_{x,y} \,\}.$$

A Figura 3 ilustra o resultado dessas etapas.

2.2.2. Fechamento Morfológico

Para corrigir partes desconexas de um mesmo caractere — que poderiam ser interpretadas como caracteres distintos pelo algoritmo de segmentação — aplicou-se fechamento morfológico, operação que preenche descontinuidades estreitas e pequenos buracos [2, p. 420]. O fechamento é dado por $A \bullet B = (A \oplus B) \ominus B$.

Nessa etapa, as imagens foram separadas em dois grupos: o Grupo 1 reúne amostras maiores e com maior espaçamento entre caracteres; o Grupo 2 contém imagens menores e caracteres mais próximos. Assim, empregaram-se elementos estruturantes de tamanhos distintos para evitar fusões indevidas durante a dilatação $A \oplus B$.

Usou-se um *kernel* 21×21 para o grupo 1 e um *kernel* 7×7 para o grupo 2, ambos compostos por valores 1. Exemplos dos resultados podem ser vistos nas figuras 4 e 5.

2.3. Segmentação por Componentes Conexas

Após o pré-processamento, as imagens foram segmentadas usando a função `label` do módulo `measure` da biblioteca *scikit-image* [3], que rotula componentes conectadas por vizinhança-8 segundo o algoritmo de Wu et al. (2005) [4]. Em seguida, a função `regionprops` foi utilizada para obter a posição e o tamanho das *bounding boxes* que devem conter cada caractere.

Com essas informações, os símbolos foram recortados diretamente das imagens originais, preservando detalhes que poderiam ter sido degradados na etapa morfológica. Alguns exemplos desses recortes podem ser vistos na Figura 6.

2.4. Avaliação por Humanos

Das 149 amostras processadas pelo *pipeline*, foram obtidos 2530 recortes. A etapa final consistiu em uma avaliação humana: os recortes foram organizados em três grupos de cerca de 50 lotes (cada lote correspondente a uma amostra), e cada grupo foi avaliado por três voluntários, totalizando nove avaliadores.

Os participantes julgaram cada recorte como "bom" ou "ruim" segundo critérios previamente definidos. Um recorte foi considerado *bom* quando continha um caractere íntegro sem invasões de outros, quando apresentava pequenas invasões que não gerassem ambiguidade, ou quando exibisse borrões ou leve desfoque, desde

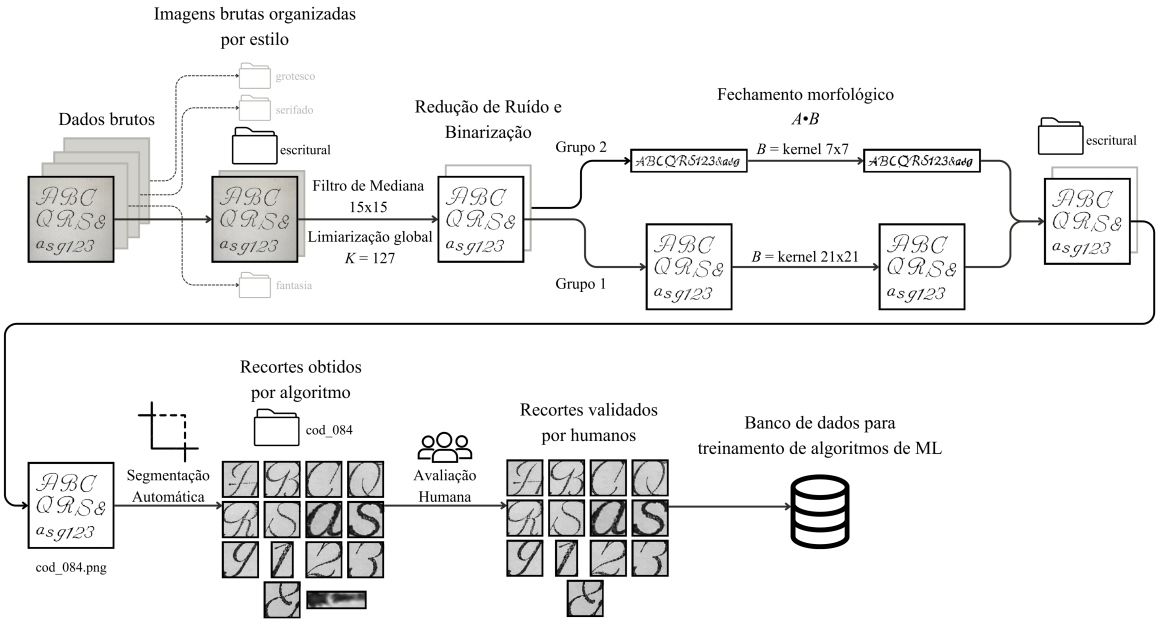

Figura 2: *Pipeline* de construção do banco de dados de amostras tipográficas.

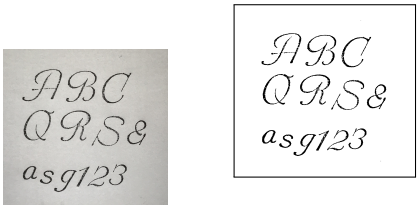

Figura 3: Imagem original (esquerda) e imagem filtrada e binarizada (direita) após processamentos. Borda adicionada para visualização dos limites.

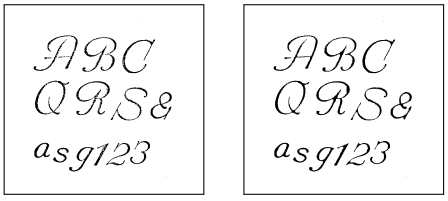

Figura 4: Imagem do Grupo 1 binarizada (esquerda) e com aplicação de fechamento morfológico (direita). Bordas adicionadas para visualização de limites.

Figura 5: Imagem do Grupo 2 binarizada (cima) e com aplicação de fechamento morfológico (baixo). Bordas adicionadas para visualização de limites.

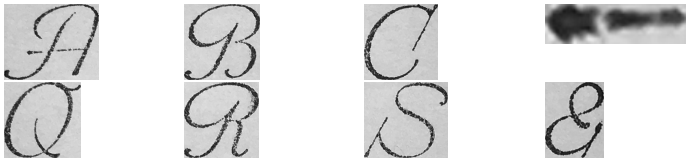

Figura 6: Alguns caracteres recortados a partir de *bounding boxes* encontradas pelo algoritmo de segmentação por componentes conexas.

que ainda permitisse reconhecer o caractere. Por outro lado, recortes foram classificados como *ruins* quando apresentavam partes soltas ou elementos irreconhecíveis, fragmentos incompletos mesmo que reconhecíveis, dois ou mais caracteres agrupados, ou ainda linhas, pontos ou outros elementos sem informação útil.

Recortes avaliados como bons por ao menos dois voluntários serão incluídos no banco de dados final. A avaliação levou cerca de 25 minutos por participante e os resultados foram registrados em um arquivo `CSV`. O banco de dados consolidado será produzido a partir dessas respostas.

3. Impactos da Pesquisa

O principal impacto esperado desta pesquisa é a agilização do processo de catalogação de tipos móveis em famílias de fontes tipográficas no Acervo Tércio Gau-

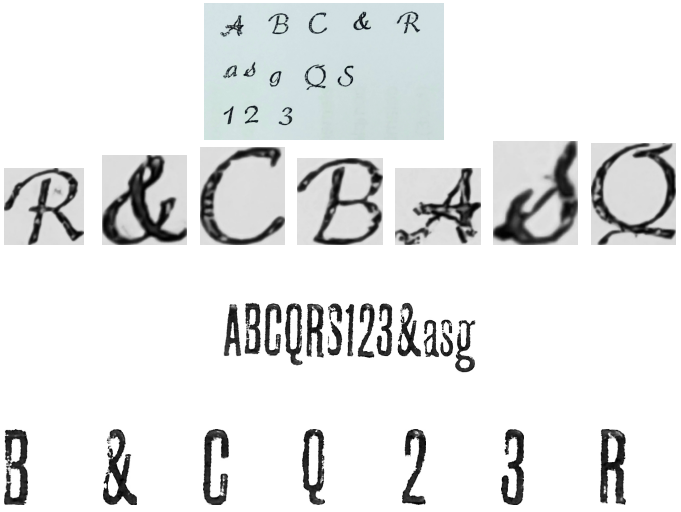

Figura 7: Alguns recortes considerados "bons" originados de cod_002 e cod_088.

dêncio do Museu Paulista, por meio de modelos de visão computacional capazes de reconhecer rapidamente a família ao qual o tipo pertence. Futuramente, essa base de dados poderá possibilitar novas classificações e análises de grande relevância para a pesquisa museológica, como a identificação da origem e época de fabricação dos tipos móveis, o contexto de uso das fontes e o reconhecimento tipográfico em livros impressos, entre outras potenciais aplicações.

4. Resultados e Discussão

Como ainda não há resultados quantitativos sobre a precisão dos recortes produzidos pelo *pipeline*, a análise limita-se à inspeção visual. As figuras 7 e 8 mostram quatro exemplos de segmentações. Na primeira imagem, o algoritmo apresenta bom desempenho, gerando recortes majoritariamente adequados segundo os critérios da seção 2.4. Nos segunda imagem, porém, o resultado é insatisfatório.

Observe que, nos dois primeiros exemplos, os recortes exibem caracteres inteiros, sem invasões. Nos dois últimos, porém, há grande quantidade de recortes imprecisos: no terceiro, predominam fragmentos de caracteres; no quarto, surgem recortes contendo múltiplos caracteres.

Como o algoritmo de segmentação identifica componentes conectadas por vizinhança-8, esses dois últimos casos ilustram seu comportamento quando os caracteres apresentam, respectivamente, partes desconexas devido a traços interrompidos na imagem original, ou quando estão excessivamente justapostos ou levemente sobrepostos. Nessas situações, parte dos recortes torna-se inadequada para compor o banco de dados final, por não atender aos critérios de integridade estabelecidos, prejudicando sua utilidade na consolidação do conjunto.

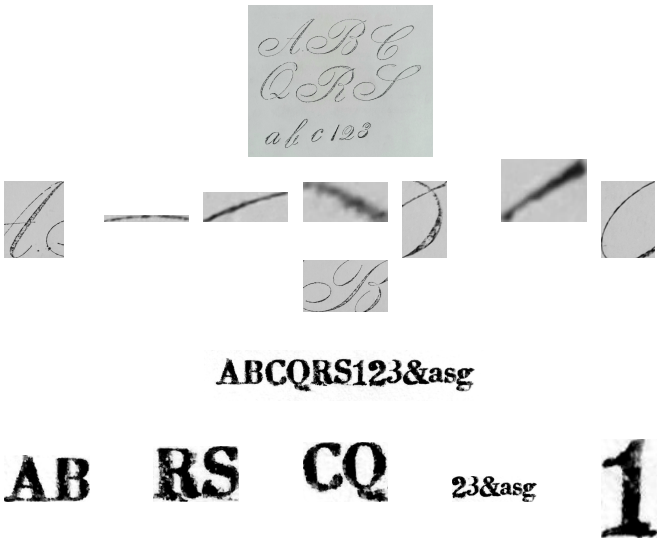

Figura 8: Alguns recortes considerados "ruins" originados de cod_016 e cod_147.

5. Conclusão

Com base no desenvolvimento realizado e nos resultados qualitativos obtidos, conclui-se que o *pipeline* atual é eficaz na geração de caracteres segmentados para o banco de dados quando as imagens de entrada apresentam símbolos bem distribuídos, permitindo inclusive o uso de técnicas de reconstrução — como fechamento ou dilatação com *kernels* maiores — para conectar componentes fragmentadas sem fundir objetos distintos. Embora os métodos empregados apresentem limitações, eles atendem à proposta de exploração do problema, especialmente na construção do banco de dados; e, caso sejam fornecidas amostras que satisfaçam os requisitos de distribuição espacial, espera-se melhor desempenho do algoritmo e a formação de um conjunto mais robusto, diverso e composto por um maior número de exemplares válidos e bem segmentados.

Agradecimentos

Agradecemos à equipe do Museu Paulista pelo apoio ao projeto, em particular ao Prof. Dr. Fabio Mariano Cruz Pereira, e ao colega de laboratório MSc. Vagner Inacio de Oliveira.

O presente trabalho foi realizado com apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Brasil. Processo n° 2024/23738-0.

Referências

[1] F. Bruneli, Y. Ohashi, F. Pereira, F. Figueroa, and S. Lima, "Gavetas tipográficas da coleção Tércio Ferdinando Gaudêncio: procedimentos para catalogação das famílias tipográficas," *Anais do XV Congresso Brasileiro de Pesquisa e Desenvolvimento em Design - P&D Design*, 2024.

[2] R. C. Gonzalez and R. E. Woods, *Processamento de Imagens Digitais*, 3rd ed. Pearson Prentice Hall, 2010.

[3] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: Image processing in python," *CoRR*, vol. abs/1407.6245, 2014. [Online]. Available: http://arxiv.org/abs/1407.6245

[4] K. Wu, E. Otoo, and A. Shoshani, "Optimizing connected component labeling algorithms," https://escholarship.org/uc/item/7jg5d1zn, Lawrence Berkeley National Laboratory, 2005.