

Sistema modular para otimização de parâmetros de funções de reconhecimento de padrões utilizando inteligência artificial

Edson Cordeiro do Valle^{a,*}

^a Laboratório de Visão Computacional Aplicada à Indústria (LICV), Departamento de Engenharia Química, Universidade Federal do Rio Grande do Sul, RS, Brasil. * edsoncv@ufrgs.br

RESUMO

Entre os conceitos e ferramentas utilizados para automação na Indústria 4.0 estão o uso de metrologia por imagens e visão computacional. Através de técnicas e algoritmos de análises de imagens digitais é possível realizar caracterizações quantitativas de parâmetros geométricos baseado em dados extraídos dessas imagens, porém, tais técnicas necessitam da configuração de diversos parâmetros para serem executados, gerando um problema combinatório complexo. Nesse contexto o objetivo do presente trabalho é apresentar um sistema para seleção e configuração de filtros para detecção, quantificação, e medição de círculos, visando a futura caracterização de bolhas de processos de flotação, utilizando um agente de inteligência artificial conhecido como Aprendizado por Reforço, ou, *Reinforcement Learning* (RI). No sistema proposto, três imagens com 900 círculos foram utilizadas para a detecção, com posição e raios dos círculos conhecidos. A estratégia de aprendizado por reforço utilizando Cadeias de Decisão de Markov foi implementada para o sistema de detecção de círculos. Como métrica a da qualidade do resultado, foi utilizado o número de círculos encontrados, a razão entre a interseção e a união das áreas dos círculos (gabarito e encontrado) e a diferença da Distância Euclideana entre os centros dos círculos (gabarito e encontrado). O sistema foi implementado em linguagem Java, utilizando o pacote de funcionalidades de tratamento e processamento de imagens de código aberto *OpenCV*. Além do sistema de detecção de círculos, foram testados alguns filtros para pré-tratamento das imagens. Em seguida o algoritmo de detecção de círculos com os filtros de pré-tratamento proposto foi aplicado, testando parâmetros da estratégia de busca das Cadeias de Markov. O sistema encontrou os parâmetros otimizados dos filtros de forma bastante eficiente, selecionando os melhores filtros a serem aplicados, indicando a necessidade de 1 a 3 filtros para encontrar os 900 círculos nas 3 imagens, sem a presença de falsos positivos. Como conclusões, a metodologia proposta apresentou erros de posicionamento e magnitude dos raios reduzidos, validando a metodologia de uso da técnica de Aprendizado por Reforço com Cadeias de Decisão de Markov.

Palavras-chave: Análise de imagens, aprendizagem por reforço. *reinforcement learning*, detecção de círculos.

1 Introdução

No processo atual de migração tecnológica das indústrias que buscam a digitalização e automação industrial, conhecido como Indústria 4.0, entra como ator importante o uso de análises de imagens digitais de produtos e matérias-primas para inspeção e controle de qualidade tanto nos processos fabris quanto nos laboratórios (Lins et al., 2025). Em um processo evolutivo das ferramentas de análise de imagens digitais, a Visão Computacional, ou *Computer Vision*, usa ferramentas avançadas de computação, como Inteligência Artificial, para análises e interpretações mais avançadas dessas imagens e vídeos (Marinho et al., 2026). Através da análise de imagens digitais e da aplicação de algoritmos de Visão Computacional é possível realizar, por exemplo, a identificação de formas geométricas, a medição de altura e área interfaces, a medição de quantidade de partículas, entre outras.

Com a grande amplitude de capacidade dos métodos de análise de imagens digitais e da Visão Computacional, existe a possibilidade de substituição e automação de ensaios analíticos de laboratórios utilizando tais técnicas. Entretanto, para a extração de dados quantitativos de imagens digitais é necessário que as imagens passem por uma cadeia de tratamento, denominados filtros de imagens, que necessitam de configuração de seus parâmetros para serem executados, sendo, a maioria, de natureza contínua. Considerando que, muitas vezes, 2 ou mais filtros precisam ser aplicados sequencialmente com o intuito de realizar a caracterização quantitativa de imagens, encontrar o melhor conjunto de parâmetros gera um problema combinatório bastante complexo.

Nesse contexto, o objetivo do presente trabalho é apresentar um sistema seleção e configuração de parâmetros de filtros de imagens digitais para detecção, quantificação, e medição de círculos, visando a futura caracteriza-

ção de bolhas de processos de flotação por ar dissolvido, utilizando um agente de inteligência artificial conhecido como Aprendizado por Reforço, ou, *Reinforcement Learning* (RI).

2 Metodologia

2.1 Seleção da metodologia, métodos e ambiente de desenvolvimento

Postulada a complexidade relativa à determinação de parâmetros de filtros para pré-tratamento de imagens, foi iniciada a elaboração de um sistema baseado em *software*, que, a partir de um sistema de Inteligência Artificial, denominado Aprendizado por Reforço, ou *Reinforcement Learning* (RI) fosse capaz tanto de ajustar os parâmetros dos filtros de pré-processamento quanto os parâmetros da função para detecção de círculos. Para a implementação do sistema de RI, foi selecionada a linguagem de programação Java como linguagem de desenvolvimento devido à sua praticidade e de recursos avançados de programação como Programação Orientada à Objetos.

Na etapa seguinte, foi selecionado o pacote (ou biblioteca) computacional para o tratamento de imagens e detecção de círculos. No sentido de atender os objetivos, foi selecionada a biblioteca de funções OpenCV (Bradski, 2000), pois, possui uma grande variedade de funções para leitura escrita e tratamento de imagens. Para a detecção de círculos, a função utilizada foi a *Hough Circles*, que detecta os círculos das imagens utilizando o método de Círculos de Hough ou Transformada Hough. A função da Transformada de Hough possui seis parâmetros, que devem ser ajustados para encontrar apropriadamente o número de círculos em uma determinada imagem. Antes da execução da função da Transformada de Hough, é recomendado aplicar filtros de pré-tratamento, como candidatos iniciais, foram utilizados os filtros: Brilho-Contraste, Desfoque Gaussiano, Equalização de Histograma por Contraste Adaptativo (CLAHE), Bilateral e Limiar (*Threshold*) Adaptativo.

2.2 Metodologia para avaliação dos filtros e estratégias de detecção de círculos

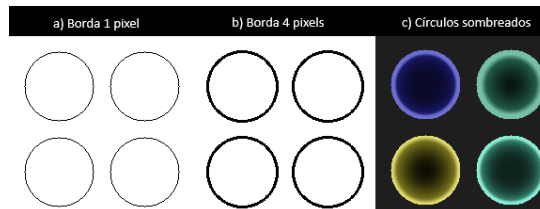
Nessa etapa, foi selecionada uma metodologia para a escolha dos parâmetros dos filtros e da Transformada de Hough. Entre as metodologias elencadas inicialmente, estava a variação combinatória simples dos parâmetros, também denominado problema de ajuste de hiperparâmetros. Em função da presença de 18 parâmetros (6 da função de Círculos de Hough, 12 dos filtros de pré-processamento), seria gerado um problema combinatório relativamente complexo, assim, foi pesquisada na literatura, algoritmos mais eficientes para tal avaliação. Bergstra e Bengio, 2012, sugerem uma forma semi-aleatória para a busca de parâmetros, evitando a busca exaustiva. Autores como Snoek et al., 2012, sugerem uma metodologia de Otimização Bayesiana para ajuste de hiperparâmetros. Visando uma abordagem diferenciada em relação ao sugerido pela literatura, foi selecionado um método alternativo de Aprendizado por Reforço (RI), ou *reinforcement learning* (RL) (Li, 2023). A aprendizagem por reforço é um tipo de aprendizagem de máquina, ou *machine learning* (ML), em que um agente "aprende" o comportamento ideal por meio da interação com o ambiente. Na RL o agente "aprende" por tentativa e erro, recebendo respostas na forma de recompensas ou penalidades por suas ações. Na prática, de uma maneira resumida, são gerados círculos com posição e raios conhecidos, ou gabarito, e o agente explora os parâmetros dos filtros e do método da Transformada de Hough que maximizem encontrar esses círculos com a posição e os raios mais próximos dos originais. Esse tipo de busca é realizada de forma metodológica, sendo apresentada a seguir.

2.2.1 Visão geral do método

No sistema proposto por RL, três imagens com um conjunto de 900 círculos cada foram geradas, a primeira com a espessura média de bordas dos círculos de 1 *pixel*, a segunda com espessura média de bordas dos círculos de 4 *pixels* e a terceira imagem é constituída de círculos com bordas sombreadas simulando bolhas com sombreamento lateral, sendo a cor e espessura do sombreamento variando aleatoriamente. A posição dos centros e raios conhecidos foram geradas programaticamente, gerando uma imagem com dimensões de 2745 por 2745 *pixels*. Os raios dos círculos gerados foram variados entre 13 e 38 *pixels* e o afastamento mínimo entre bordas foi de 15 *pixels*. A matriz com os dados de posição do centro e dos raios foi armazenada, sendo denominada "gabarito". Uma amostra com somente 4 círculos das 3 imagens são apresentadas na Figura 1.

Durante a execução do algoritmo nas imagens descritas anteriormente, são aplicados os filtros de pré-processamento, seguido da detecção pela Transformada de Hough. Após a detecção dos círculos, baseado nos critérios apresentados anteriormente, o agente recebe uma nota, semelhante à avaliação de uma função objetivo de um problema de otimização. Quanto maior for essa nota, mais próximo ao gabarito o agente se encontra,

Figura 1: Amostra das 3 figuras utilizadas para detecção de círculos. a) Círculos com espessura de 1 *pixel*. b) Círculos com espessura de 4 *pixels*. c) Círculos sombreados simulando imagens de bolhas.



reforçando a busca em regiões dos parâmetros em que essa nota se mantenha maximizada. Na implementação realizada, foi estipulada uma restrição no espaço de busca dos parâmetros, assim como o passo de variação dentro dessa margem.

2.2.2 Modelagem do Problema como Processo de Decisão de Markov (MDP)

O processo de otimização dos parâmetros dos filtros de pré-processamento e dos parâmetros da Transformada de Hough foi modelado como um Processo de Decisão de Markov (MDP) finito (Bellman, 1957). Na tomada de decisão para escolha dos parâmetros o agente autônomo interage com o ambiente de detecção, aprendendo uma política ótima para incrementar ou decrementar os parâmetros iterativamente, maximizando uma função de recompensa baseada em um gabarito (*Ground Truth*) conhecido. A função de recompensa aborda os seguinte fatores quando compara os círculos encontrados com o gabarito:

- Encontrar o círculo corretamente;
- Não detectar falsos positivos (encontrar um círculo onde não existe);
- Não encontrar mais círculos que os iniciais;
- Acertar a posição do centro do círculo S_{dist} dentro de uma tolerância (usada a distância Euclideana entre gabarito e encontrado);
- Acertar a razão entre Interseção e União do círculo (IoU) encontrado (indicador para acerto do raio e indretamente da posição).

Também foi implementada uma Métrica Híbrida Ponderada (*Dense Reward Shaping*, Ng et al., 1999), promovendo a aproximação dos círculos detectados em direção aos centros corretos. A qualidade de cada círculo detectado c_{det} em relação a um círculo real, c_{real} , é calculada por:

$$Score_{hibrido} = (W_{IoU} \cdot IoU) + (W_{dist} \cdot S_{dist}) \quad (1)$$

em que W_{IoU} e W_{dist} são pesos configuráveis e seus somatórios devem ser igual a unidade. O índice IoU varia entre 0 e 1, sendo 1 para um acerto perfeito em que as 2 áreas, de sobreposição e união, são idênticas. O índice de distância Euclideana S_{dist} também é normalizado no intervalo de 0 a 1 baseado na máxima distância do raio aceita, D_{max} , usada para ponderar o acerto do centro do círculo:

$$S_{dist} = 1 - \left(\frac{\sqrt{(x_{det} - x_{gt})^2 + (y_{det} - y_{gt})^2}}{D_{max}} \right) \quad (2)$$

em que x_{det} e y_{det} é a posição dos centros, x e y, detectadas e x_{gt} e y_{gt} é a posição dos centros, x e y, dos gabaritos. O algoritmo proposto será guiado pela média do $Score_{hibrido}$, definida como:

$$MScore_{hibrido} = W_{IoU} \cdot mIoU + W_{dist} \cdot mS_{dist} \quad (3)$$

em que $mIoU$ e mS_{dist} são as médias aritméticas dos IoU e S_{dist} para todos os círculos, respectivamente. O número parâmetros dos filtros de pré-processamento e da Transformada de Hough explorados pelo método proposto

Tabela 1: Faixa de exploração dos parâmetros

Parâmetro	Número de combinações
Brilho-Contraste	546
Desfoque Gaussiano	40
CLAHE	112
Bilaeral	16641
Limiar (<i>Threshold</i>) Adaptativo	525
Transformada de Hough	9199008

são apresentados na Tabela 1. Após a seleção da linguagem de programação, do método de RI, dos critérios da função de recompensa e da região de busca dos parâmetros, foi iniciada a elaboração do programa, sendo utilizada a ferramenta Gemini Plus de Inteligência Artificial para elaboração do código fonte em Java. No *prompt* da ferramenta foram inseridos os critérios desejado na arquitetura, como modularidade, uso de recursos de programação orientada à objetos (herança, polimorfismo e padrões de projeto) e paralelização, sendo esta última, inserida posteriormente, para fins de melhoria de tempo computacional. O programa desenvolvido foi disponibilizado no repositório GitHub no seguinte sítio: <https://github.com/EDSONCV/CircleTests/> sob licença Apache-2.0.

2.2.3 Experimentos Computacionais

O método proposto foi implementado em Linguagem Java e executada em um computador Intel(R) Core(TM) i7-14700T (1.30 GHz) com 32,0 GB de memória em um sistema operacional Windows 11. O referido computador possui 20 núcleos e 28 processadores lógicos. Em cada uma das 3 imagens previamente descritas, foram avaliados: melhores filtros individuais; conjunto de todos os filtros em série; tempo computacional com 27 *threads* e influência dos parâmetros de peso, W_{IoU} e W_{dist} . Os resultados principais do método são apresentados na seção seguinte.

3 Resultados

Além das métricas previamente apresentadas, também são apresentadas nas tabelas de resultados uma métrica mais intuitiva, denominada $mDist$, que denota a média aritmética da Distância Euclidiana entre os círculos gabaritos e os círculos encontrados. Nessa métrica, quanto menor seu valor, na média, mais próximo estão os círculos detectados dos círculos gabaritos.

Os resultados para a imagem com círculos com bordas finas são apresentados na Tabela 2, sendo que a detecção de círculos sem pré-tratamento não identificou círculos na imagem, possivelmente devido à espessura das bordas dos círculos, sendo esse resultado não apresentado na Tabela 2 para redução de espaço. Na imagem de círculos com bordas finas é possível observar que a adição do filtro de Desfoque Gaussiano (DG) apresenta bons resultados, encontrando todos os círculos dispostos no gabarito. Considerando um elevado número de círculos, o índice $mIoU$ acima de 0,95 como resultado, para fins de comparação com outros métodos de literatura, é considerado um valor de excelência (Lin et al., 2014). Considerando também a métrica $mDist$ para os resultados encontrados menor que 1, denominada precisão de *subpixel*, também é considerado um resultado de excelência em metrologia de alta exatidão baseada em modelos que detectam bordas, como a Transformada de Hough utilizada (Sun et al., 2021). Como métrica secundária de resultado final de cada configuração, o valor da função objetivo também pode ser comparado, sendo que a mesma considera, além das métricas individuais de cada círculo, penalizações por encontrar um número superior ou inferior de círculos. A influência dos pesos W_{IoU} e W_{dist} também não foram muito significativas nos resultados entre os mesmos filtros de pré-processamento. Já a adição de todos os filtros na cadeia acabou por piorar os resultados, encontrando um número maior ou menor de círculos conforme a combinação dos pesos W_{IoU} e W_{dist} , sendo isso indicado pelo valor inferior dos índices (menor $mIoU$ e maior $mDist$), assim como da função objetivo.

Os resultados para a imagem com círculos com bordas espessas são apresentados na Tabela 3. Através dos dados da referida tabela, é possível observar que o sistema de detecção de círculos sem pré-tratamento conseguiu identificar círculos na mesma, mas com uma eficiência reduzida quando comparada com conjuntos de filtros de pré-processamento, destacando uma maior magnitude da métrica $mDist$ acima de 1 *pixel*. Além disso, é possível

Tabela 2: Resultado da metodologia com filtros de bordas finas.

Condição de Pré-Processamento	Configuração Inicial $W_{IoU} - W_{dist}$	Círculos Encontrados	Função Objetivo	$mIoU$	$mDist$ (px)	$MScore_{hibrido}$	Tempo (s)
DG*	0,1-0,9	900	42581	0,9716	0,3	0,9463	158
DG*	0,5-0,5	900	42850	0,9749	0,4	0,9522	401
DG*	0,9-0,1	900	42033	0,9749	0,4	0,9341	662
Todos Filtros	0,1-0,9	902	38858	0,952	0,7	0,8628	358
Todos Filtros	0,5-0,5	806	30983	0,7765	0,9	0,8535	253
Todos Filtros	0,9-0,1	905	40371	0,9194	1,5	0,8971	294

*Desfoque Gaussiano

Tabela 3: Resultado da metodologia com filtros de bordas espessas.

Condição de Pré-Processamento	Configuração Inicial $W_{IoU} - W_{dist}$	Círculos Encontrados	Função Objetivo	$mIoU$	$mDist$	$MScore_{hibrido}$	Tempo (s)
Sem Filtro	0,1-0,9	901	32475	0,9192	1,5	0,7225	454
Sem Filtro	0,5-0,5	900	36413	0,9209	1,5	0,8092	154
Sem Filtro	0,9-0,1	900	40435	0,9209	1,5	0,8986	313
DG*	0,1-0,9	900	42529	0,9599	0,3	0,9451	319
DG*	0,5-0,5	900	40728	0,9528	0,7	0,9051	290
DG*	0,9-0,1	900	40728	0,9528	0,7	0,9051	229
Todos Filtros	0,1-0,9	901	37797	0,9351	0,9	0,8392	377
Todos Filtros	0,5-0,5	900	39341	0,93	0,9	0,8743	680
Todos Filtros	0,9-0,1	900	41167	0,9324	1,2	0,9148	762

*Desfoque Gaussiano

observar que a adição do filtro de pré-processamento de Desfoque Gaussiano (DG) apresenta bons resultados, quando comparado com a detecção de círculos sem a aplicação de filtros. Quando comparados com a mesma imagem e o mesmo filtro de pré-processamento, é observado uma pequena diferença, no caso superior, para o conjunto de W_{IoU} de 0,1. Já a adição de todos os filtros na cadeia de pré-processamento não trouxe resultados tão inferiores quanto ao caso da imagem de bordas finas, possivelmente, pelo fato de que as bordas da imagem são mais espessas, sendo identificados todos os círculos, mas com eficiência menor quando comparados com os demais resultados da imagem de bordas espessas.

Os resultados para a imagem com círculos com bordas sombreadas são apresentados na Tabela 4. Nesse caso o sistema de detecção de círculos sem pré-tratamento não conseguiu identificar círculos, por esse motivo, esses dados foram omitidos da Tabela 4. Também é possível observar que, pela característica da imagem, o filtro de DG deve ser aplicado como última etapa na cadeia de pré-processamento. Tal estratégia foi realizada manualmente, observando quantitativamente os resultados de algumas combinações preliminares. Esse fato faz parte da complexidade desse tipo de problema da seleção de filtros de pré-processamento, no sentido de que a ordem da cadeia de filtros e a adição ou remoção de filtros altera de forma substancial os resultados desta etapa. Os círculos na referida imagem passaram a ser identificados com a aplicação da cadeia de pré-processamento com o filtro Bilateral, seguido do filtro TA e DG. Na aplicação com todos os filtros à imagem de círculos sombreados os resultados foram inferiores quando comparados com as cadeias de filtros anteriores, encontrando um número superior de círculos em algumas combinações de pesos $W_{IoU} - W_{dist}$, conforme apresentado nas duas últimas linhas da Tabela 4. De uma maneira geral, através das análises das tabelas 2, 3 e 4, para todas as imagens, quando comparados dentro da mesma cadeia de aplicação de filtros de pré-processamento, houve pouca variabilidade dos resultados ao variar os pesos $W_{IoU} - W_{dist}$.

Tabela 4: Resultado da metodologia com círculos com bordas sombreadas.

Condição de Pré-Processamento	Configuração Inicial $W_{IoU} - W_{dist}$	Círculos Encontrados	Função Objetivo	$mIoU$	$mDist$ (pixel)	$MScore_{hibrido}$	Tempo (s)
Bilateral+TA**+DG	0,1-0,9	900	42331	0,926	0,3	0,9407	1037
Bilateral+TA**+DG	0,5-0,5	900	42990	0,9687	0,3	0,9553	679
Bilateral+TA**+DG	0,9-0,1	900	43456	0,9682	0,3	0,9657	1517
Todos Filtros	0,1-0,9	900	42351	0,9268	0,3	0,9411	2230
Todos Filtros	0,5-0,5	909	41599	0,9128	0,3	0,9254	797
Todos Filtros	0,9-0,1	901	41364	0,9167	0,3	0,9186	1313

**Threshold Adaptativo

4 Conclusões

Considerando a aplicação da metodologia nas três imagens, foi observado que os pesos $W_{IoU} - W_{dist}$ apresentam pouca influência nos índices de detecção dentro do mesmo conjunto de filtros de pré-processamento. De um modo geral, foi possível observar a adequação da técnica de uso de agente de Inteligência Artificial de RI na escolha de parâmetros dos filtros de pré-tratamento e da função da Transformada de Hough.

A escolha da arquitetura do sistema com a utilização da linguagem de programação Java, utilizando os recursos de Programação Orientada à Objetos e Padrões de Projeto, propiciaram a elaboração de um sistema modular em relação à adição de filtros e paralelização que otimizaram o tempo de busca. O uso de sistemas de Modelos de Processamento de Linguagem Natural, como o Gemini Pro, para elaboração do código-fonte, agilizou a elaboração e implementação do método, embora, pela experiência ao longo da implementação, o autor não sugere o uso desse tipo de sistema sem conhecimentos prévios de programação e sem um conhecimento teórico preliminar, pois, ao longo do processo de testes e elaboração, diversas correções básicas foram necessárias.

Para o caso da imagem de círculos com bordas sombreadas, devido à complexidade da mesma, foi necessário realizar a escolha e ordenamento preliminar da cadeia de filtros de pré-processamento de uma forma visual e qualitativa, sendo essa uma das limitações do método implementado, sendo atribuída como uma sugestão para trabalhos futuros uma maneira automática de realizar a seleção e ordenamentos dos filtros mais relevantes.

Referências

- Bellman, R. (1957). A Markovian decision process. *Journal of mathematics and mechanics*, 6(5), 679–684.
- Bergstra, J., & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*, 13(10), 281–305. <https://jmlr.org/papers/v13/bergstra12a.html>
- Bradski, G. (2000). The OpenCV Library. *Dr. Dobbs's Journal of Software Tools*.
- Li, S. E. (2023). Introduction to Reinforcement Learning. Em *Reinforcement Learning for Sequential Decision and Optimal Control* (pp. 1–14). Springer Nature Singapore. https://doi.org/10.1007/978-981-19-7784-8_1
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *European conference on computer vision*, 740–755.
- Lins, R. G., Santos, R. E., & Gaspar, R. (2025). Vision-Based Measurement for Quality Control Inspection Integrated Into a Die-Casting Process in Industry 4.0 Era. *IEEE Access*.
- Marinho, F. d. O., Zomer, T. T. d. S., Durão, L. F. C. d. S., Cauchick-Miguel, P. A., & Zancul, E. (2026). Computer vision in Quality 4.0: empirical insights from industrial demands. *Production*, 36, e20250096. <https://doi.org/10.1590/0103-6513.20250096>
- Ng, A. Y., Harada, D., & Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. *ICML*, 99, 278–287.
- Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian Optimization of Machine Learning Algorithms. *Advances in Neural Information Processing Systems 25 (NeurIPS 2012)*. <https://proceedings.neurips.cc/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf>
- Sun, H., et al. (2021). A Novel Subpixel Circle Detection Method Based on the Blurred Edge Model. *IEEE Transactions on Instrumentation and Measurement*, 71, 1–11.