

# **Estimativa da Porcentagem de Músculo e Tecido Conjuntivo em Imagens com Machine Learning**

**Antônio Eduardo de Oliveira Carmo, Eduardo Paz Vieira, Paulo H. A. de Andrade**

<sup>1</sup>Ciência da Computação – Universidade Federal Rural do Semi-Árido (UFERSA)  
Mossoró – RN – Brasil

{antonio.carmo, eduardo.vieira, paulo.andrade}@alunos.ufersa.edu.br

**Resumo.** Este estudo propõe o uso de um modelo de aprendizado de máquina para a classificação e quantificação de tecidos musculares e conjuntivos em imagens histológicas de implantes tratados com plasma atmosférico a frio, visando superar as limitações da análise manual tradicional. A abordagem convencional, que utiliza pontos dispersos nas imagens e análise por especialistas, apresenta imprecisões devido ao número reduzido de amostras e à potencial ocultação de áreas relevantes. O modelo desenvolvido classifica pixels individualmente, permitindo uma análise mais detalhada e precisa. Os resultados mostraram um coeficiente  $R^2$  de 0,908 na fase de validação com especialistas, demonstrando alta fidelidade das respostas. Além disso, o tempo de processamento foi reduzido significativamente, com cada imagem sendo analisada em menos de 0,15 segundos.

## **1. Introdução**

Durante um estudo veterinário sobre o efeito da osseointegração de implantes tratados com plasma atmosférico a frio no tecido peri-implantar em ratos, houve a necessidade de contabilizar a densidade muscular e de tecido conjuntivo em amostras coletadas por imagens, através de fotomicrografias dos cortes histológicos em microscópio de luz. O processo de contagem é popularmente feito através da dispersão de pontos espalhados pela imagem para que um especialista julgue se naquele ponto há um músculo, tecido conjuntivo, ou alguma outra classe, neste caso, com o auxílio do software Image Pro Plus.

Neste estudo os especialistas optaram por usar 54 pontos, onde para cada quantidade  $k$  de pontos classificados como músculos, a quantidade de pontos classificados como tecido será  $54 - k$ . Dessa forma, para cada valor de  $k$ , temos uma combinação única de pontos. Isso infere que o número de combinações diferentes é simplesmente o número de valores possíveis para  $k$ , ou seja, 55.

Essa limitação causa uma imprecisão nos resultados, pois imagens com contagens bem diferentes podem acusar valores exatamente iguais de densidade muscular. Além disso, a disposição dos pontos pode esconder áreas da imagem com potencial de alterar consideravelmente o resultado.

Atualmente, algoritmos de aprendizado de máquina têm se mostrado muito relevantes em análises de imagens por sua precisão e velocidade de resposta [Óscar Lorente and Rana 2021]. Dessa forma, este trabalho se propõe a implementar um modelo capaz de classificar individualmente *pixels* de imagens e realizar a contagem de

cada classe presente. Além disso, para expandir ainda mais o potencial da análise, a solução deve ser capaz de identificar classes anômalas que não se encaixam como músculo ou tecido conjuntivo.

## 2. Estudo e Preparação do *Dataset*

O banco de imagens fornecido pelos especialistas contém 225 imagens diferentes. A análise das imagens mostrou que não há necessidade de um pré-processamento, visto que já faz parte da rotina de captura das imagens colocar pigmentações para realçar as diferenças entre as classes.

Dessa forma, listou-se 159 imagens (cerca de 70% do *dataset*) de forma aleatória para garantir que haja representatividade dentre as amostras, o que melhora a generalização dos modelos. Estas imagens foram rotuladas como treinamento para não serem repetidas no processo de validação.

## 3. Requisitos dos Modelos

A escolha dos modelos se restringiu aos não supervisionados, visto que, por mais que as classes sejam bem definidas, seria inviável realizar uma validação humana pixel a pixel de cada imagem para obter a função de perda e as métricas de acuracidade durante a fase de treino.

Além disso, alguns algoritmos como redes neurais ou K-nearest Neighbors (KNN) seriam inviáveis por exigir uma cama de input e dimensionalidade igual ao total de *pixels* [Tripathi 2021] [Guo et al. 2004], o que traria um modelo exageradamente pesado e impossibilitaria a fácil adaptação a diferentes tamanhos de imagem. De forma semelhante, os algoritmos BIRCH e *Agglomerative Clustering* também foram descartados por apresentar problemas com excesso de memória alocada.

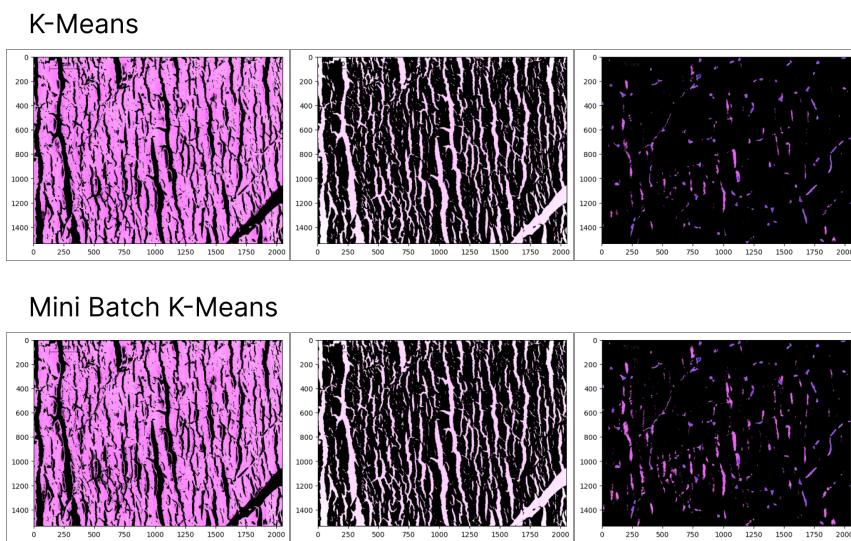
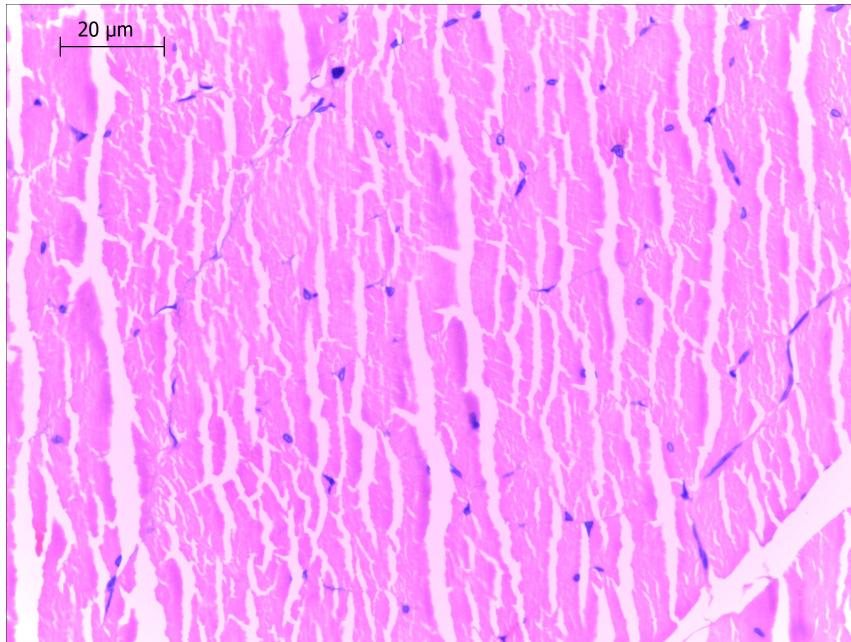
Por fim, o modelo deve ser capaz de receber como *input* a quantidade de classes ou *clusters*, inviabilizando o uso do DBSCAN e do algoritmo *Affinity Propagation*. Com isso, os algoritmos estudados, que cumprem os requisitos listados acima, foram: K-means e Mini Batch K-means.

### 3.1. Implementação do modelo

Para os algoritmos configurou-se apenas os hiper-parâmetros da quantidade de *clusters* e o valor de *random state* para maior controle da distribuição dos centróides [Sklearn K-means documentation ].

Visto que a técnica de clusterização incrementa sua complexidade com a quantidade de *pixels* analisados, para o processo de treinamento, todas as imagens foram redimensionadas para uma resolução de 512 x 512. Dessa forma, cada imagem ficou composta por 262.144 pixels, cada um tendo os 3 valores da escala RGB, totalizando 786.432 diferentes entradas por imagem.

Iterando os valores do *random state*, determinou-se que o que melhor representava as classes analisadas foi o 1. Na Figura 1 pode-se notar exemplos da classificação de cada classe isolada juntamente com a imagem original.



**Figure 1. Comparação entre as classificações de músculo, tecido conjuntivo e anomalias, respectivamente, dos modelos K-means e Mini Batch K-means.**

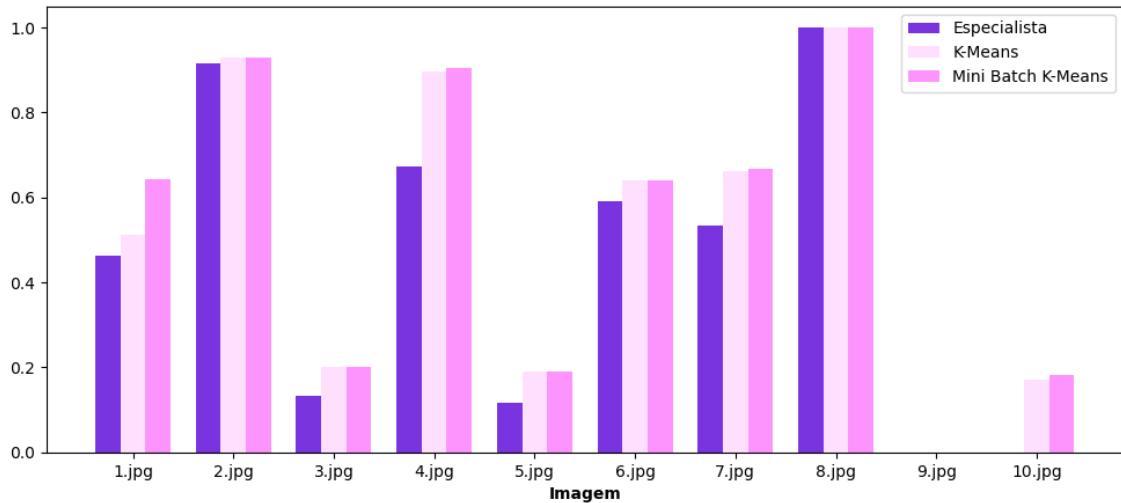
#### 4. Validação

O processo de validação foi conduzido usando uma pequena amostragem do *dataset*, visto que esta etapa exigiu uma minuciosa análise por parte do especialista.

No total, 10 imagens de resolução 32 x 32 foram geradas a partir de cortes aleatórios em imagens randômicas do *dataset* de validação. Cada imagem foi levada a um software de edição de imagens para que o especialista delimitasse os tecidos conjuntivos através de uma ferramenta de seleção livre. A área selecionada foi então pintada de preto, e imagem exportada. O último passo consistiu em contar, através de um script, quantos *pixels* pretos haviam na imagem editada.

## 5. Resultados

Após expor as imagens de validação ao modelo, os resultados foram comparados com os do especialista, como ilustra a Figura 2. Como métrica, adotou-se o  $R^2$ , que totalizou em 0,908 para o modelo K-means e em 0,876 para o Mini Batch K-means, mostrando uma ótima fidelidade à análise do especialista.



**Figure 2. Validação dos modelos em relação à porcentagem de tecido conjuntivo.**

Além disso, considerou-se também o tempo de processamento de cada modelo, separando entre tempo de treinamento e tempo de inferência. O teste consistiu em uma média de 5 execuções da rotina de treinamento e classificação de 20 imagens inteiras, sendo usadas a mesma amostragem do *dataset* para os dois algoritmos. Como mostrado na Tabela 1, a diferença de tempo indicou o modelo K-means como o mais apropriado.

Modelo	Tempo de treinamento	Tempo de inferência	Total
K-means	30, 60s	2, 66s	33, 26s
Mini Batch K-means	78, 80s	2, 58s	71, 38s

**Table 1. Comparação entre os tempos de processamento de cada modelo**

Com isso, a seleção final do modelo favoreceu o algoritmo de K-means tradicional por sua melhor fidelidade às previsões de validação e menor tempo de processamento.

Estima-se que, para gerar uma classificação completa de uma imagem sem cortes ou redimensionamento, esta implementação demora menos de 0,15 segundos, em média. Esta tarefa, quando feito manualmente, exige alguns minutos de dedicação exclusiva por parte do especialista para cada imagem. Além disso, há muito mais confiabilidade em uma abordagem *pixel a pixel*, pois esta, ao invés de se limitar à combinação de alguns pontos, estende a quantidade de possíveis respostas para uma combinação entre todos os *pixels* da imagem.

### 5.1. Aplicação

Para encapsular o modelo e dar uma melhor experiência para os usuários do sistema, implementou-se uma aplicação web de simples interface e uso. A página principal dá

ao cliente a opção de importar uma ou mais imagens que serão classificadas. Após a classificação, uma área para análise individual de cada imagem é exibida, trazendo a imagem junto a um gráfico da porcentagem de cada classe. Por fim, há a opção de exportar os resultados analisados para um arquivo *CSV*.

## 6. Conclusão

Tendo obtido uma ótima conformidade no processo de classificação das 3 classes, este trabalho cumpriu todos os objetivos inicialmente impostos. Resgatando o problema que motivou a implementação, fica evidente a eficiência em termos de tempo e acurácia do algoritmo K-means para a análise das fotomicrografias.

Futuros trabalhos nesta área podem envolver melhorias no processo de detecção das anomalias e em uma generalização das classes para se adaptar a diferentes cenários de análise microscópicas por imagens.

## References

- Guo, G., Wang, H., Bell, D., and Bi, Y. (2004). Knn model-based approach in classification.
- Sklearn K-means documentation. <https://scikit-learn.org/1.5/modules/generated/sklearn.cluster.KMeans.html>. Accessed: 2024-10-07.
- Tripathi, M. (2021). Analysis of convolutional neural network based image classification techniques. *Journal of Innovative Image Processing (JIIP)*, pages 100–117.
- Òscar Lorente, I. R. and Rana, A. (2021). Image classification with classic and deep learning techniques.