

# Meta-Characteristics Extraction from Image Datasets for Selection of Convolutional Neural Networks

Lucas V. Dias<sup>1</sup>, Péricles B.C. Miranda<sup>1</sup>, André C. A. Nascimento<sup>1</sup>,  
Filipe R. Cordeiro<sup>1</sup>, Rafael Ferreira Mello<sup>1</sup>, Ricardo B. C. Prudêncio<sup>2</sup>,  
Ricardo D. Oliveira<sup>1</sup>

<sup>1</sup> Departamento de Computação,  
Universidade Federal Rural de Pernambuco (UFRPE)

<sup>2</sup> Centro de Informática – Universidade Federal de Pernambuco  
Recife – PE – Brazil

{pericles.miranda}@ufrpe.br

**Abstract.** *Convolutional Neural Networks (CNNs) is the main solution for image classification tasks in different applications. However, selecting the most suitable CNN and its parameters for a given image dataset is usually performed by trial and error, which may take much time and computational cost. This paper proposes the Dataset2Vec method and employs Meta-Learning (MtL) to select CNN architectures for image classification. Dataset2Vec adopts a deep neural network to extract features from images datasets, embedding them in a single feature vector. To evaluate the proposed solution, it was adopted to select among six CNN algorithms for 45 two-classes image datasets. The results showed that the MtL using Dataset2Vec outperformed by different baseline methods in all performance measures evaluated, indicating the proposal was able to extract representative features from datasets of images for CNN selection.*

**Resumo.** *As Redes Neurais Convolucionais (RNCs) são a principal solução para a tarefa de classificação de imagens em diferentes aplicações. No entanto, a seleção da RNC mais adequada e de seus parâmetros para uma determinada base de imagens geralmente é realizada por tentativa e erro, o que pode levar muito tempo e custo computacional. Este artigo propõe o método Dataset2Vec e emprega o método de Meta-Aprendizado (MtA) para selecionar arquiteturas de RNCs para classificação de imagens. O Dataset2Vec adota uma rede neural profunda para extrair características de bases de imagens, incorporando-as em um único vetor. Para avaliar a solução proposta, adotou-se a seleção entre seis algoritmos da RNC para 45 bases de imagens de classificação binária. Os resultados mostraram que o MtA usando o Dataset2Vec superou os diferentes métodos baseline em todas as medidas de desempenho avaliadas, indicando que a proposta foi capaz de extrair recursos representativos dos conjuntos de dados de imagens para seleção de RNCs.*

## 1. Introdução

As Redes Neurais Convolucionais (RNCs) provaram ser a principal solução para tarefas relacionadas à classificação de imagens [Szegedy et al. 2015]. No entanto, o bom desempenho de uma RNC depende de sua arquitetura, seus parâmetros, e das características do

problema de classificação a ser resolvido [Wang et al. 2017]. Definir qual RNC é mais adequada para um determinado problema não é trivial. Até mesmo um especialista pode levar muito tempo e custo computacional para avaliar diferentes arquiteturas de RNC até que a melhor seja escolhida [Wang et al. 2017]. A tarefa de selecionar a RNC mais apropriada para um determinado problema pode ser considerado um problema de seleção de algoritmos.

Meta-Aprendizado (MtA) é um paradigma de aprendizado de máquina que trata o problema de seleção de algoritmos como uma tarefa de aprendizado supervisionado [Miranda et al. 2014]. Cada exemplo de treinamento para MtA (meta-exemplo) armazena as características de um problema passado (no nosso caso, uma base de imagens) e informações sobre o desempenho obtido por um algoritmo (no nosso caso, uma arquitetura de RNC) no problema. Este conjunto de meta-exemplos forma uma base de meta-conhecimento chamada meta-base. O MtA recebe como entrada a meta-base para aprender um meta-modelo para prever o algoritmo mais adequado para um novo problema com base nas suas características. Em nosso contexto, uma vez que o meta-modelo é construído, os algoritmos de RNC podem ser sugeridos para novos problemas sem a necessidade de avaliar empiricamente diferentes RNCs candidatas (conforme realizado usando técnicas de tentativa e erro). Apesar de seus resultados promissores em diferentes aplicações, o sucesso do MtA depende fortemente da etapa de extração de meta-características [Chao et al. 2020]. Se as características extraídas do problema não o descreverem adequadamente, o conhecimento adquirido pelo meta-modelo será prejudicado [Lorena et al. 2018]. Um número extenso de descritores foi proposto para extrair características de imagens [Yong et al. 2005, Takemoto and Yokota 2009, Wang et al. 2017, Achille et al. 2019]. No entanto, apenas características, tais como, intensidade, cores, textura, borda, entre outras, são usadas para descrever bases de imagens.

Neste trabalho, a ideia de extrair automaticamente uma representação vetorial de bases de imagem é estendida. Este artigo propõe o método Dataset2Vec, cujo objetivo é criar um *embedding* de características extraídas de uma dada base de imagens. O Dataset2Vec implementa um algoritmo de aprendizagem profundo pré-treinado para extrair as características de cada imagem da base. Os *embeddings* de cada imagem são agregados, formando um único *embedding* resultante, que representa as características da base de imagens como um todo. A qualidade do Dataset2Vec, em extrair características de bases de imagens, foi avaliada em um estudo de caso de MtA. Foram considerados seis algoritmos de RNC e 45 bases de imagens binárias criadas a partir do CIFAR-10. Para isso, um conjunto de meta-exemplos foi fornecido como entrada para o MtA, e a precisão na seleção do algoritmo RNC mais apropriado para o conjunto de dados de entrada foi avaliada. Para fins de comparação, consideramos três *baselines*: (1) *OneR*, que sempre recomenda um algoritmo RNC exclusivo; (2) *Random*, que fornece recomendações aleatórias de algoritmo; e (3) *Ground Truth* (GT), que sempre recomenda o melhor algoritmo (o *ground-truth*) encontrado anteriormente em cada conjunto de dados. O método GT é um limite superior de comparação e, na prática, exige que todos os algoritmos candidatos sejam avaliados empiricamente em cada conjunto de dados. Os resultados experimentais mostraram que o meta-modelo, usando o Dataset2Vec, superou os dois primeiros *baselines*. Além disso, o meta-modelo alcançou um desempenho estatisticamente equivalente ao método GT, o que significa que ele seleciona o melhor algoritmo de RNC para cada base de imagens ou, pelo menos, seleciona um algoritmo candidato próximo do ótimo.

Este artigo está estruturado da seguinte forma. A seção 2 apresenta os trabalhos relacionados. Na seção 3 são apresentadas as perguntas de pesquisa do presente trabalho. A solução proposta é abordada na Seção 4. A Seção 5 apresenta os experimentos realizados e analisa os resultados alcançados. Finalmente, conclusões e trabalhos futuros são discutidos na Seção 6.

## 2. Trabalhos relacionados

A classificação de imagens através de RNCs, se tornou muito popular nos últimos tempos. No entanto, um grande número de hiper-parâmetros e aspectos arquiteturais para uma determinada tarefa de classificação de imagens motivaram o uso do MtA para ajudar os pesquisadores a encontrar a RNC mais apropriada para uma tarefa em questão. Motivado por isso, o MtA tem sido aplicado em alguns problemas, incluindo tarefas em visão computacional, por exemplo, segmentação, classificação, entre outras. Em Aguiar et al. [Aguiar et al. 2019] foi proposto o uso de MtA para selecionar o algoritmo de segmentação de imagem mais adequado para uma determinada imagem. Neste trabalho, foram adotados descritores tradicionais, que extraem cor, intensidade, borda, entre outros. Essas características físicas da imagem também são conhecidas como *handcrafted*, e se mostraram relevantes na descrição de imagens.

Em [Wang et al. 2017] os autores propuseram um sistema de recomendação de modelos de RNC para tarefas de classificação de imagens. O recomendador mediu a complexidade, juntamente com os descritores de imagem SURF [Bay et al. 2006] para estimar a capacidade de classificação dos modelos de RNC pré-treinados. Sendo assim, uma pontuação de complexidade é atribuída a cada imagem nos dados de treinamento, para descrever a dificuldade de um classificador classificá-la corretamente e uma média é calculada para representar o conjunto de dados como um todo. Eles consideraram um simples classificador de centróide para avaliar a dificuldade de classificação, usando um único ponto-chave (centro de imagem) para calcular o descritor SURF para cada imagem e calculando a média de todos os vetores em cada classe para representá-lo. Os resultados demonstram que a abordagem proposta pode recomendar o modelo ideal de RNC em sete conjuntos de dados de classificação de imagens.

Outra área que vem recebendo muita atenção nos últimos anos é a tarefa de aprender meta-características que podem representar a tarefa de aprendizado (por exemplo, as características do conjunto de dados) sem calcular explicitamente um conjunto de descritores pré-definidos. A ideia é semelhante à tarefa de aprender representações vetoriais de palavras (por exemplo, word2vec, GloVe etc.) em tarefas de Processamento de Linguagem Natural (PLN). Achille et al. [Achille et al. 2019] propõem uma abordagem intimamente relacionada ao nosso problema, chamada TASK2VEC. Os autores propõem um método para caracterizar a semelhança de tarefas (ou seja, conjunto de dados) através do uso de um tipo de uma rede neural *auto-encoder*, chamada "probe network". Essa rede calcula um *embedding* com base em estimativas da matriz de informações de Fisher associada aos seus parâmetros. Os *embeddings* produzidos são independentes de características como distribuição de classe e semântica de rótulo de classe. Os resultados mostraram que o método proposto pode ajudar na seleção de um extrator de características para uma nova tarefa. Nossa proposta amplia essa ideia com um processo diferente para construção do *embedding*, além de usá-lo para recomendar ativamente as melhores arquiteturas de RNC para um novo problema de classificação de imagem.

### 3. Pergunta de Pesquisa

Este artigo propõe duas contribuições para a literatura. Primeira, uso de um algoritmo de MtA para recomendar arquiteturas de RNC para uma nova base de imagens. Para fazer isso, propomos um extrator de características, baseado em redes *auto-encoder* para encontrar a representação vetorial (*embedding*) mais apropriada para uma dada base de classificação de imagens. Segunda, o artigo apresenta experimentos que mostram que as características extraídas das RNCs podem ser eficientemente agregadas para atuar como poderosas meta-características em tarefas de seleção de algoritmos, usando MtA, para classificação de imagens.

**PERGUNTA DE PESQUISA:** *As características extraídas pelo Dataset2Vec descrevem bem o problema a ponto de contribuir com a recomendação do MtA?*

### 4. Dataset2Vec para Seleção de Algoritmos

Este trabalho propõe um novo método para extrair características para representar uma base de imagens. A representação derivada é adotada no MtA como meta-características para auxiliar a seleção do algoritmo de RNC para uma dada base de imagens.

#### 4.1. Dataset2Vec

O processo de extração de características de problemas de aprendizagem tem um papel fundamental no processo de recomendação de algoritmos. No contexto deste trabalho, cada problema se trata de uma base composta por uma variedade de imagens. Este trabalho propõe um método para caracterizar uma base de imagens, de forma a fornecer uma descrição unificada e representativa de todas as imagens contidas na base. O Algoritmo 1 mostra cada passo do método proposto.

---

**Algoritmo 1:** Pseudo-código do Dataset2Vec.

---

```
1: Input: image_dataset
2: Output: final_embedding
3:
4: all_embeddings = []

6: for image in image_dataset do
7:     embedding = feature_extraction(image)
8:     all_embeddings.append(embedding)
9: end
10: final_embedding = embedding_aggregation(all_embeddings)
11: return final_embedding
```

---

Dada uma base de imagens (*image\_dataset*) como entrada, a primeira etapa (linha 6-9) extrai um vetor de características de cada imagem (*embedding*). O vetor de características de todas as imagens (*all\_embeddings*) é fornecido como entrada para a etapa de síntese de características (linha 10). Nesta etapa, é feita a agregação de todos os vetores de características em um único, que representa toda a base de imagens.

Neste trabalho, a função *feature\_extraction* foi implementada com um modelo RNC pré-treinado conhecido como MnasNet (mais detalhes em [Tan et al. 2019]). É importante mencionar que nove arquiteturas diferentes de RNC foram testadas: DenseNet, Inception v.3, MnasNet, MobileNet, ResNet, ResNext, ShuffleNet, SqueezeNet e Wide Residual Network (WRN). Quando comparado às outras abordagens, o MnasNet extraiu mais características descritivas das imagens, melhorando o desempenho do Dataset2Cec para a tarefa de seleção de algoritmos. A MnasNet é uma rede construída de forma otimizada usando uma abordagem de *Automated Neural Architecture Search* para projetar modelos usando o aprendizado por reforço (mais detalhes em [Zoph et al. 2018]). Para o nosso problema, usamos um modelo MnasNet pré-treinado no Imagenet, por meio de *transfer learning*. Os parâmetros pré-treinados são usados para extrair as características das bases de imagens adotadas aqui. É importante mencionar que o vetor de características de cada imagem é formado a partir da extração de características realizadas pelas camadas convolucionais que precedem a camada totalmente conectada da RNC.

A função *embedding\_aggregation* recebe como entrada o vetor de características de cada imagem do conjunto de dados (*all\_embeddings*). Este módulo usa o algoritmo de agrupamento *K-means* para agrupar os vetores de características em  $K$  grupos. O *K-means* usou a distância Euclidiana e foi executado dez vezes. Se o conjunto de dados de imagens tiver  $C$  classes, então  $K = C$ . Depois de agrupar todos os vetores, serão retornados  $K$  centróides, um por classe. A saída (vetor de características resultante) é a média entre os  $K$  centróides.

A representação vetorial do conjunto de dados obtida pode ser vista como um *embedding* em um espaço denso de alta dimensão. Além de que, ele também não depende diretamente dos rótulos das imagens, mas apenas de uma representação holística (por exemplo, aplicação de várias camadas convolucionais) das imagens que compõem a base. O *embedding* de cada base de imagens será usado como meta-características para o MtA. Como tal, espera-se que a distância entre dois *embeddings* de duas bases de classificação de imagem semelhantes também estejam mais próximas no espaço de *embeddings*. É importante destacar que criamos um método flexível, no qual as funções *feature\_extraction* e *embedding\_aggregation* podem ser implementadas de acordo com o especialista.

## 4.2. Procedimento do MtA

O MtA possui duas fases: (1) a modelagem do MtA, que explora o conhecimento adquirido anteriormente em tarefas semelhantes; e (2) a etapa de recomendação. Na modelagem do MtA, os algoritmos candidatos de RNC são avaliados em bases de imagens e diferentes medidas de avaliação são computadas. Além disso, cada base de imagens tem suas meta-características extraídas. A etapa de avaliação define os "meta-alvos", ou seja, as classes/rótulos a serem previstos pelo sistema de recomendação do MtA (no caso deste trabalho, algoritmos de RNC). Esses dois processos geram uma meta-base. Finalmente, dada a meta-base de treinamento como entrada, algoritmos de classificação (meta-aprendizes) são aplicados para induzir um meta-modelo. O meta-modelo mapeia a relação entre as meta-características e o meta-alvo. Na fase de recomendação do MtA, o meta-modelo pode ser aplicado a uma nova base de imagens, que é representada pelos valores de suas meta-características extraídos. O meta-modelo retorna o algoritmo de RNC mais adequado para essa base de imagens. As próximas seções descrevem em detalhes cada componente apresentado.

### 4.2.1. Meta-base

Para a criação da meta-base, usamos o conjunto de dados CIFAR-10 [Krizhevsky and Hinton 2009], que é comumente adotado para avaliar modelos de RNC. Esse conjunto de dados é composto por 60.000 imagens de treinamento 32x32 coloridas, rotuladas em 10 categorias. Para os experimentos, 45 conjuntos de dados foram criados a partir do CIFAR-10. Para isso, as imagens de cada classe foram separadas, criando assim dez bases de dados diferentes com uma classe. Em seguida, foram realizadas todas as combinações simples possíveis de bases de dados ( $C_{10,2}$ ), totalizando 45 bases de imagens com duas classes. Cada uma das 45 bases de dados será representada como um meta-exemplo na meta-base a partir das suas meta-características e meta-alvo. Mais detalhes são mostrados a seguir.

### 4.2.2. Meta-características

O conjunto de medidas usadas para extrair as características relevantes de um dado problema é chamado de meta-características. No nosso caso, produzimos um vetor numérico para cada imagem na base (consulte a 4.1). As camadas convolutivas de uma RNC funcionam como filtros para extrair características visuais, como bordas, contornos e outros. A partir de várias características simples, a rede aprende a criar filtros mais sofisticados, capazes de extrair características mais complexas. No final das camadas convolutivas de uma rede RNC, possui um conjunto de características visuais, espaço latente ou *embedding*. Todas as características aprendidas nas camadas convolutivas que precedem a camada totalmente conectada formam as características da imagem de entrada. Neste trabalho, o vetor de cada imagem possui 1281 valores de características, obtido pela aplicação das camadas convolucionais do MnasNet. Como todos as bases de imagens têm duas classes, a função *embedding\_aggregation* usou  $K = 2$  para produzir o vetor resultante de características que representa a base de imagens inteira. Este vetor de características irá compor o meta-exemplo.

### 4.2.3. Meta-alvos

O valor do meta-alvo indica qual algoritmo de RNC candidato é o mais adequado para um determinado conjunto de dados de imagens. Como meta-alvos, consideramos as seguintes arquiteturas de RNC: AlexNet1, MiniGoogLeNet1, MiniVGG1, MiniVGG3, ResNet10 e SmallSqueezeNet. Os algoritmos de RNC foram avaliados quanto à acurácia (*Accuracy*), revocação (*recall*), precisão (*precision*) e medida-F (*F-Score*). Cada medida foi dimensionada como uma pontuação contínua entre 0 e 1. Como estamos lidando com diferentes medidas, usamos um método de tomada de decisão com vários critérios chamado *Technique for Order of Preference by Similarity to Ideal Solution* (TOPSIS) para ranquear os algoritmos [Olson 2004]. O algoritmo de melhor classificação se torna o valor do rótulo (meta-alvo). Embora seis meta-alvos tenham sido considerados para a criação de meta-base, apenas quatro meta-alvos obtiveram desempenho suficiente para serem atribuídos aos meta-exemplos: AlexNet, MiniGoogleNet1, MiniVGG1 e MiniVGG3. Estes algoritmos são meta-alvo em 6%, 28,8%, 13,34% e 51,86% dos meta-exemplos da meta-base, respectivamente.

#### 4.2.4. Meta-aprendiz

O algoritmo de classificação *Random Forest* (RF) foi utilizado como meta-aprendiz. Esse método foi escolhido porque pode induzir modelos com diferentes vieses de aprendizado e está bem estabelecido na academia e na indústria, sendo utilizado com sucesso em múltiplas tarefas preditivas [Fernández-Delgado et al. 2014]. O RF foi implementado usando a linguagem Python e o pacote Scikit-Learn [Pedregosa et al. 2011], com valores padrão de hiper-parâmetro. O desempenho preditivo do meta-aprendiz foi avaliado usando a estratégia *Leave-One-Out Cross Validation* (LOO-CV).

Também foram usados diferentes *baselines* nas comparações experimentais: um modelo que recomenda sempre uma classe exclusiva para todo o conjunto de dados (OneR) e um modelo que fornece recomendações aleatórias (Random). Além disso, usamos um limite superior como o *ground-truth* (GT), que conhece o algoritmo RNC que obteve o melhor desempenho de classificação.

### 5. Resultados experimentais

No experimento realizado, comparamos o desempenho preditivo de diferentes modelos: um meta-modelo usando as características extraídas pelo *Dataset2Vec*, e modelos *OneR* e *Random*. Os desempenhos preditivos obtidos por cada modelo podem ser vistos na Tabela 1. Cada linha da tabela representa os resultados de média e desvio padrão que um meta-modelo alcançou para cada medida de desempenho diferente, considerando todos as 45 bases de imagens usadas neste trabalho. Para fins de comparação, os resultados do GT também foram incluídos na tabela.

Como pode ser visto, os resultados alcançados pelo *Dataset2Vec* superaram os modelos *OneR* e *Random* em todas as métricas de desempenho. Além disso, os resultados obtidos pelo *Dataset2Vec* também atingiram valores muito próximos aos do GT.

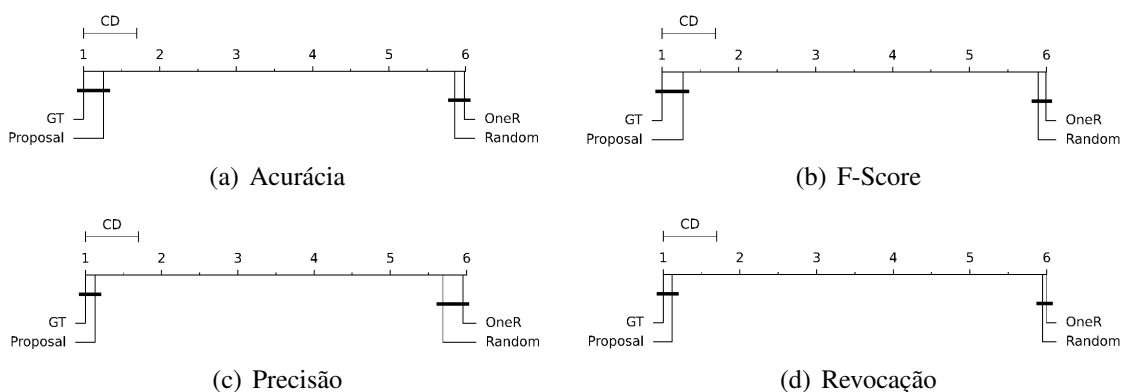
**Tabela 1. Desempenho alcançado pelos diferentes meta-modelos.**

Meta-modelo	Desempenho (Média & Desvio Padrão)			
	Acurácia	<i>F-Score</i>	Precisão	Revocação
<i>Dataset2Vec</i>	94,32 ( $\pm$ 0,03)	94,34 ( $\pm$ 0,03)	91,94 ( $\pm$ 0,05)	94,97 ( $\pm$ 0,04)
<i>OneR</i>	91,18 ( $\pm$ 0,05)	90,99 ( $\pm$ 0,05)	88,28 ( $\pm$ 0,06)	91,43 ( $\pm$ 0,05)
<i>Random</i>	91,95 ( $\pm$ 0,05)	91,86 ( $\pm$ 0,05)	89,16 ( $\pm$ 0,06)	92,39 ( $\pm$ 0,05)
GT	94,69 ( $\pm$ 0,03)	94,74 ( $\pm$ 0,03)	92,30 ( $\pm$ 0,05)	95,40 ( $\pm$ 0,03)

Para a avaliação estatística, adotamos o teste de Mann-Whitney considerando um nível de significância de  $\alpha = 0,05$ . Este teste é usado para comparar as diferenças entre dois grupos independentes. Assim, empregamos uma comparação pareada entre todas as abordagens, incluindo o GT. A hipótese nula significa que a recomendação retornada pelas abordagens comparadas é estatisticamente igual. Caso a hipótese nula seja rejeitada, as recomendações são consideradas estatisticamente diferentes. As células destacadas

(em cinza mais escuro), na Tabela 1, são consideradas estatisticamente iguais entre si. Em outras palavras, o modelo *Dataset2Vec* alcançou resultados estatisticamente iguais aos do GT em todas as medidas de desempenho. Por outro lado, os resultados obtidos pelas abordagens *OneR* e *Random* não rejeitaram a hipótese nula quando comparados ao *Dataset2Vec* e GT, sendo superados por elas em todas as medidas de desempenho.

Visando uma análise estatística mais aprofundada, também realizamos o teste *post hoc* de Nemenyi. Este teste pretende identificar grupos de dados que divergem após um teste estatístico de múltiplas comparações (como o teste de Mann-Whitney). Os grupos são formados de acordo com o desempenho das abordagens envolvidas. O desempenho de duas abordagens é significativamente distinto se as classificações médias correspondentes diferirem em pelo menos um valor de diferença crítica (CD). Para melhor visualização dos resultados da comparação, usamos uma representação gráfica chamada de diagrama de diferença crítica [Demšar 2006].



**Figura 1. Comparação do desempenho alcançado por cada meta-modelo na recomendação de RNCs de acordo com o teste de Nemenyi. Meta-modelos que não são significativamente diferentes ( $\alpha = 0,05$  e  $CD = 0,69$ ) são agrupados.**

Como se pode ver, o modelo *Dataset2Vec* é agrupado com o GT em todas as quatro medidas de desempenho, corroborando com a conclusão anterior. Os meta-modelos *Random* e *OneR* apresentaram desempenho semelhante entre si em relação à acurácia, *f-score*, e revocação. Este resultado mostra que o modelo *Dataset2Vec* pode recomendar meta-alvos adequados para o problema de entrada. É importante mencionar que, ao considerarmos neste trabalho, uma meta-base de 45 exemplos, com apenas quatro meta-alvos, aumenta as chances de o *OneR* e o *Random* recomendarem o melhor algoritmo corretamente por sorte. Dessa forma, a diferença nos resultados entre meta-modelos e estes dois meta-modelos tende a ser cada vez maior à medida que a meta-base cresce. Faz parte de nossos objetivos futuros aumentar o meta-base em números de meta-exemplos e meta-alvos.

## 6. Conclusão

Este artigo propõe o método *Dataset2Vec*, que é um novo método para extrair características em uma representação vetorial de comprimento fixo *embedding* de uma base de imagens. O intuito é extrair características mais representativas de uma base de imagens para identificar melhor um algoritmo ou classes de algoritmos capazes de executar tarefas de aprendizado com mais eficiência. Para criar a meta-base, foram considerados seis



meta-alvos, e foram consideradas 45 bases de imagem, totalizando 45 meta-exemplos. Para fins experimentais, foi utilizado o procedimento de Meta-Aprendizado (MtA), seu módulo de extração de características foi configurado com o método proposto, e comparado com diferentes métodos *baseline*. Os resultados mostraram que o MtA usando o *Dataset2Vec* superou os diferentes *baselines* em todas as medidas de desempenho avaliadas, sendo estatisticamente igual ao *ground truth*.

Como trabalho futuro, pretendemos aumentar o número de meta-exemplos, assim como diversificar e equilibrar o número de meta-alvos na meta-base. Também devemos testar ainda mais a robustez e as limitações dos *embeddings* em bases mais extensas e diversas, bem como estratégias diferentes para agregar os *embeddings* de várias imagens de uma base. Além disso, objetivamos comparar o método proposto com descritores de imagem estado da arte no âmbito de seleção de algoritmos, e verificar quais das abordagens apresenta maior precisão na recomendação.

## Referências

- Achille, A., Lam, M., Tewari, R., Ravichandran, A., Maji, S., Fowlkes, C., Soatto, S., and Perona, P. (2019). Task2Vec: Task Embedding for Meta-Learning.
- Aguiar, G. J., Mantovani, R. G., Mastelini, S. M., de Carvalho, A. C., Campos, G. F., and Junior, S. B. (2019). A meta-learning approach for selecting image segmentation algorithm. *Pattern Recognition Letters*, 128:480–487.
- Bay, H., Tuytelaars, T., and Van Gool, L. (2006). Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer.
- Chao, W.-L., Ye, H.-J., Zhan, D.-C., Campbell, M., and Weinberger, K. Q. (2020). Revisiting Meta-Learning as Supervised Learning.
- Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30.
- Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15:3133–3181.
- Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, Citeseer.
- Lorena, A. C., Maciel, A. I., de Miranda, P. B., Costa, I. G., and Prudêncio, R. B. (2018). Data complexity meta-features for regression problems. *Machine Learning*, 107(1):209–246.
- Miranda, P. B., Prudêncio, R. B., De Carvalho, A. P., and Soares, C. (2014). A hybrid meta-learning architecture for multi-objective optimization of svm parameters. *Neuro-computing*, 143:27–43.
- Olson, D. L. (2004). Comparison of weights in topos models. *Mathematical and Computer Modelling*, 40(7-8):721–727.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830.

- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9.
- Takemoto, S. and Yokota, H. (2009). Algorithm selection for intracellular image segmentation based on region similarity. In *2009 Ninth International Conference on Intelligent Systems Design and Applications*, pages 1413–1418. IEEE.
- Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., and Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2820–2828.
- Wang, S., Sun, L., Fan, W., Sun, J., Naoi, S., Shirahata, K., Fukagai, T., Tomita, Y., Ike, A., and Hashimoto, T. (2017). An automated cnn recommendation system for image classification tasks. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 283–288. IEEE.
- Yong, X., Feng, D., Rongchun, Z., and Petrou, M. (2005). Learning-based algorithm selection for image segmentation. *Pattern Recognition Letters*, 26(8):1059–1068.
- Zoph, B., Vasudevan, V., Shlens, J., and Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.