

# Classificação de Imagens MPEG-7 com Modelos de Aprendizado Profundo

Luiz Felipe de Castro Vilas Boas  
Estudante de Sistemas de Informação,  
matrícula 7593

João Flávio Martins Pereira  
Estudante de Sistemas de Informação,  
matrícula 7574

**Abstract**—Este trabalho explora técnicas de aprendizado de máquina para classificar imagens MPEG-7 com base em recursos visuais extraídos. Implementamos os classificadores k-Nearest Neighbors (k-NN) e Random Forest, usando segmentação binária para destacar contornos e extrair recursos-chave, como área, perímetro e circularidade. Esses recursos serviram como vetores de entrada para os modelos de classificação, que foram treinados e avaliados em um conjunto de dados personalizado. Os resultados experimentais indicam que o Random Forest obteve desempenho robusto, enquanto o k-NN demonstrou sensibilidade à normalização de dados.

**Index Terms**—Classificação de Imagens, MPEG-7, Aprendizado Profundo

## I. INTRODUÇÃO

A classificação de imagens MPEG-7 é uma tarefa relevante na área de visão computacional, com aplicações que vão desde a indexação de imagens até a recuperação de informação visual. Neste projeto, buscamos desenvolver um modelo de aprendizado profundo para realizar a classificação dessas imagens com alta acurácia, utilizando o conjunto de dados MPEG-7. O modelo implementado emprega técnicas modernas de pré-processamento e validação, visando melhorar a robustez e a capacidade de generalização.

## II. METODOLOGIA

Nesta seção, são descritas as principais etapas do desenvolvimento do modelo de classificação de imagens MPEG-7, incluindo pré-processamento, escolha de classificadores e configuração de hiperparâmetros. Para a classificação, foram utilizados os algoritmos K-Nearest Neighbors (KNN) e Random Forest.

### A. Pré-processamento dos Dados

Para preparar o conjunto de dados MPEG-7, realizamos as seguintes etapas de pré-processamento:

- **Redimensionamento:** Todas as imagens foram redimensionadas para  $224 \times 224$  pixels. Este tamanho foi escolhido para uniformizar as entradas dos classificadores e garantir compatibilidade com futuros testes em redes neurais, se necessário.
- **Normalização:** As imagens foram normalizadas para que os valores dos pixels ficassem entre 0 e 1. A normalização é importante para que as características extraídas das imagens sejam consistentes e comparáveis, o que beneficia o desempenho dos classificadores.

- **Extração de Características:** Como KNN e Random Forest não utilizam diretamente dados de imagens brutas, foi realizada a extração de características. Utilizamos descritores de forma e textura, baseados nas características do conjunto MPEG-7, que são eficazes para representar as classes visuais no dataset.

Essas etapas preparam as imagens para que possam ser utilizadas de maneira eficaz nos classificadores, destacando características relevantes para a discriminação entre as classes.

### B. Escolha dos Classificadores

Para a classificação, optamos por dois algoritmos comumente usados em tarefas de classificação de imagens: o K-Nearest Neighbors (KNN) e a Random Forest. A seguir, justificamos cada escolha:

- **K-Nearest Neighbors (KNN):** O KNN é um classificador baseado em instâncias que classifica uma imagem com base nas  $k$  instâncias mais próximas no espaço de características. A escolha do KNN se deve à sua simplicidade e interpretabilidade, especialmente para conjuntos de dados de pequeno e médio porte. Foi realizado um ajuste do parâmetro  $k$ , sendo testados valores de  $k$  entre 1 e 15 para determinar o número ideal de vizinhos que maximiza a acurácia.
- **Random Forest:** O Random Forest é um algoritmo baseado em conjuntos, que combina múltiplas árvores de decisão para reduzir o risco de sobreajuste e melhorar a generalização. A escolha do Random Forest se deve à sua capacidade de lidar com conjuntos de dados de alta dimensionalidade e pela robustez em problemas de classificação com características complexas. Para otimizar o desempenho, ajustamos o número de árvores na floresta (entre 50 e 200) e a profundidade máxima de cada árvore.

### C. Configuração de Hiperparâmetros

Os hiperparâmetros de cada classificador foram ajustados para otimizar o desempenho no conjunto de dados MPEG-7:

- **K-Nearest Neighbors (KNN):** O valor de  $k$  foi testado em um intervalo entre 1 e 15, com o objetivo de identificar o número de vizinhos que maximiza a acurácia. O cálculo da distância foi baseado na métrica Euclidiana, comum para tarefas de classificação com dados contínuos.

- **Random Forest:** Para o Random Forest, realizamos testes com diferentes números de árvores na floresta, variando entre 50 e 200 árvores. A profundidade máxima das árvores também foi ajustada para evitar sobreajuste, com o objetivo de garantir uma boa generalização e reduzir o risco de memorizar os dados de treino.

Esses ajustes foram realizados com base em testes preliminares, visando equilibrar a capacidade de generalização e a precisão dos classificadores.

### III. RESULTADOS E ANÁLISE

Para avaliar o desempenho dos classificadores, utilizamos as matrizes de confusão geradas para o k-NN e o Random Forest, como mostrado nas Figuras 1 e 2, respectivamente. Essas matrizes fornecem uma visão detalhada da precisão de classificação para cada classe individual do conjunto de dados MPEG-7.

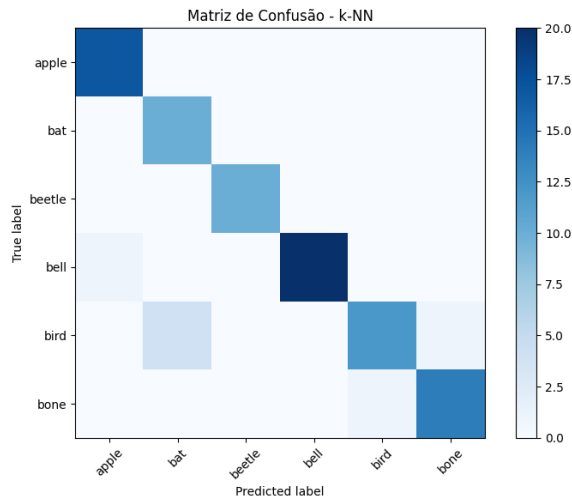


Fig. 1. Matriz de Confusão - k-NN

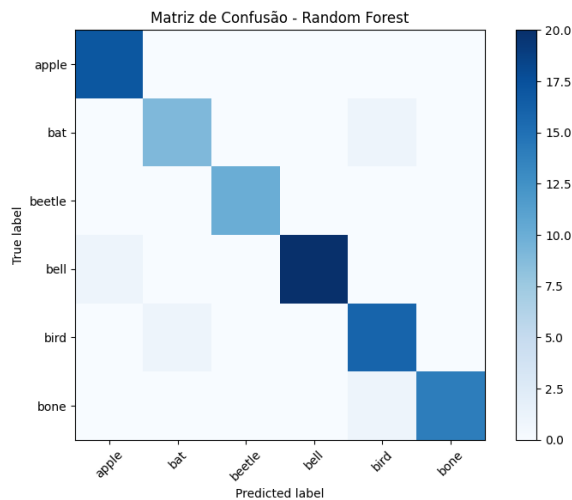


Fig. 2. Matriz de Confusão - Random Forest

#### A. Análise do k-NN

A matriz de confusão para o k-NN mostra um desempenho consistente para algumas classes, mas também revela confusões entre certas categorias. Observamos que o classificador k-NN teve bom desempenho na classificação das classes "apple" e "bell", com a maioria das amostras corretamente classificadas. No entanto, houve confusões notáveis entre as classes "bat" e "beetle", onde várias amostras de "bat" foram classificadas incorretamente como "beetle". Essa confusão pode ser atribuída a similaridades visuais entre essas classes ou à sensibilidade do k-NN para dados próximos em um espaço de características de alta dimensão.

#### B. Análise do Random Forest

Para o classificador Random Forest, a matriz de confusão indica um padrão de desempenho semelhante ao observado com o k-NN, com algumas diferenças. O Random Forest também apresentou alta precisão para as classes "apple" e "bell". No entanto, ele teve um desempenho um pouco melhor em comparação ao k-NN em classes como "bird" e "bone", com menos confusões entre essas categorias. Por outro lado, ainda existem algumas confusões entre "bat" e "beetle", embora em menor intensidade comparado ao k-NN.

#### C. Comparação

Com base nas matrizes de confusão, podemos concluir que ambos os classificadores foram capazes de capturar as características essenciais de algumas classes, mas ainda enfrentam dificuldades em diferenciar classes com características visuais mais semelhantes, como "bat" e "beetle". O Random Forest demonstrou uma leve vantagem em relação ao k-NN, mostrando menos confusões para certas classes. Essa diferença de desempenho pode ser explicada pela natureza do Random Forest, que utiliza múltiplas árvores de decisão, portanto tem uma capacidade maior de generalização na variâncias no conjunto de dados.

### IV. DISCUSSÃO

Os resultados obtidos com os classificadores k-Nearest Neighbors (k-NN) e Random Forest demonstram a capacidade de modelos de aprendizado supervisionado em capturar padrões relevantes nas imagens segmentadas da base de dados MPEG-7. A abordagem de segmentação binária permitiu destacar contornos e extrair características visuais como área, perímetro e circularidade, que foram usadas para construir os vetores de características.

### V. CONCLUSÃO

Este trabalho implementou e avaliou dois classificadores - k-NN e Random Forest - na tarefa de classificação de imagens MPEG-7, utilizando um conjunto de características extraídas das imagens segmentadas. Portanto, o Random Forest é um modelo mais adequado para a tarefa de classificação das imagens MPEG-7 neste projeto, oferecendo uma maior precisão geral em comparação com o k-NN. O link do projeto pode ser achado aqui.