



**Instituto Superior
de Engenharia**

Politécnico de Coimbra

Inteligência Computacional

Licenciatura Engenharia Informática 2025/2026

3º Ano – 1º Semestre

PROJECTO – Meta III

João Oliveira – Nº 2023144718

Diogo Ferreira - Nº 2023141377

Índice

1.	Descrição do Problema.....	3
2.	Descrição das Metodologias Utilizadas	4
3.	Arquitetura do Sistema e Aplicação Web	6
4.	Descrição da Implementação dos Algoritmos	9
5.	Análise de Resultados.....	12
6.	Conclusões.....	15
7.	Bibliografia e Referências	16

1. Descrição do Problema

A classificação automática de imagens constitui um dos problemas centrais na área da Visão por Computador, sendo amplamente utilizada em aplicações reais como sistemas de inspeção, apoio à decisão e reconhecimento visual. Neste projeto pretende-se desenvolver um sistema de classificação de imagens capaz de identificar corretamente a classe a que pertence uma imagem, recorrendo a técnicas de redes neurais convolucionais.

O problema em estudo consiste na classificação de imagens pertencentes a cinco classes distintas, sendo o conjunto de dados composto por aproximadamente 1600 imagens por classe, perfazendo um total de cerca de 8000 imagens. As imagens apresentam características visuais suficientemente distintas para permitir a aplicação de métodos de aprendizagem supervisionada, mas requerem modelos com boa capacidade de generalização para lidar com variações intra-classe.

Nas fases anteriores do projeto foi desenvolvido um modelo de classificação treinado de raiz e estudada a otimização de hiperparâmetros através de algoritmos inspirados em inteligência coletiva. No entanto, o treino de modelos convolucionais a partir do zero pode ser computacionalmente exigente e requerer grandes volumes de dados para alcançar desempenhos satisfatórios.

Neste contexto, a presente fase do projeto tem como objetivo a construção de um modelo final de classificação recorrendo a aprendizagem por transferência, utilizando uma rede neuronal convolucional pré-treinada. Esta abordagem permite aproveitar conhecimento previamente adquirido em grandes conjuntos de dados, reduzindo o custo computacional do treino e melhorando o desempenho, mesmo com conjuntos de dados de dimensão moderada.

Adicionalmente, pretende-se avaliar o impacto do número de instâncias de treino no desempenho do classificador, analisando de que forma a redução do conjunto de treino afeta a accuracy do modelo. Esta análise é particularmente relevante em cenários reais, onde a disponibilidade de dados anotados pode ser limitada.

Por fim, o modelo desenvolvido deverá ser preparado para utilização em ambiente real, sendo guardado de forma a permitir a sua integração posterior numa aplicação web destinada à classificação de novas imagens fornecidas pelo utilizador.

2. Descrição das Metodologias Utilizadas

Nas fases anteriores do projeto foi definido um modelo base de classificação de imagens e estudada a otimização de hiperparâmetros recorrendo a algoritmos inspirados em inteligência coletiva, nomeadamente o Particle Swarm Optimization (PSO). Esses algoritmos permitiram ajustar parâmetros críticos do modelo, melhorando o desempenho no conjunto de validação.

Na presente fase, essa metodologia é mantida e estendida, sendo agora aplicada no contexto de aprendizagem por transferência. Em vez de treinar uma rede convolucional de raiz, recorre-se a um modelo pré-treinado, utilizando os algoritmos swarm para otimizar apenas os hiperparâmetros das camadas finais, garantindo coerência metodológica com as fases anteriores do projeto.

2.1 Preparação e Balanceamento do Conjunto de Dados

O conjunto de dados utilizado apresenta cinco classes distintas, com um número semelhante de imagens por classe. Ainda assim, de forma a garantir uma comparação justa entre classes e evitar enviesamentos no processo de aprendizagem, foi aplicado um balanceamento do dataset por subamostragem (undersampling), limitando cada classe ao número mínimo de imagens disponível.

As imagens foram posteriormente, redimensionadas para 224×224 pixels, de acordo com os requisitos do modelo pré-treinado, normalizadas de acordo com a função `preprocess_input` da MobileNetV2, garantindo valores no intervalo [-1,1], facilitando a convergência do processo de treino.

O dataset foi dividido em três subconjuntos, Treino, utilizado para o ajuste dos pesos do modelo, Validação, utilizado durante a otimização de hiperparâmetros, Teste, reservado exclusivamente para a avaliação final do modelo em dados independentes.

2.2 Aprendizagem por Transferência

A aprendizagem por transferência foi realizada recorrendo à arquitetura MobileNetV2, pré-treinada no conjunto de dados ImageNet. Esta rede neuronal convolucional foi selecionada por apresentar um bom compromisso entre desempenho e custo computacional, sendo particularmente adequada para execução em ambientes com recursos limitados.

A camada de classificação original do modelo foi removida e a rede base foi mantida congelada, impedindo a atualização dos seus pesos durante o treino. Sobre esta base foram adicionadas novas camadas densas responsáveis pela adaptação do modelo ao

problema específico em estudo. Desta forma, o modelo beneficia das representações visuais já aprendidas, reduzindo o risco de overfitting e o tempo de treino necessário.

2.3 Otimização de Hiperparâmetros

A otimização dos hiperparâmetros foi realizada recorrendo a três abordagens distintas: dois algoritmos swarm, Particle Swarm Optimization (PSO) e Whale Swarm Algorithm (WSA), e um método de Random Search.

O Random Search foi incluído como método de referência, permitindo uma comparação direta entre algoritmos baseados em inteligência coletiva e uma abordagem aleatória de exploração do espaço de soluções.

Em todas as abordagens, foram considerados apenas hiperparâmetros associados às camadas densas finais do modelo, nomeadamente o número de neurónios da camada totalmente ligada e a taxa de aprendizagem do otimizador Adam.

Foram considerados apenas hiperparâmetros associados às camadas densas finais do modelo, nomeadamente, o número de neurónios da camada totalmente ligada, a taxa de aprendizagem do otimizador Adam.

Esta escolha permite restringir o espaço de procura, tornando o processo de otimização mais eficiente e adequado ao contexto de aprendizagem por transferência.

Durante a fase de otimização, a função de fitness foi avaliada utilizando subconjuntos fixos dos dados de treino e validação.

Esta estratégia garante consistência na avaliação das diferentes configurações de hiperparâmetros, reduzindo a variabilidade introduzida pela seleção aleatória de amostras e permitindo uma comparação mais justa entre os algoritmos PSO, WSA e Random Search.

2.4 Avaliação Experimental

Após a conclusão da fase de otimização, foi selecionada a melhor configuração de hiperparâmetros, com base no desempenho no conjunto de validação. O modelo final foi então treinado utilizando todo o conjunto de treino e avaliado num conjunto de teste independente, permitindo obter uma estimativa realista do seu desempenho.

Adicionalmente, foi realizada uma análise de sensibilidade ao tamanho do conjunto de treino, avaliando o impacto da utilização de diferentes percentagens de dados de treino no desempenho do classificador. Esta análise permite validar uma das principais vantagens da aprendizagem por transferência: a capacidade de obter bons resultados mesmo com um número reduzido de exemplos de treino.

3. Arquitetura do Sistema e Aplicação Web

3.1 Arquitetura Geral do Sistema

O sistema desenvolvido segue uma arquitetura modular, organizada de forma a separar claramente as diferentes etapas do processo de classificação de imagens. Esta abordagem facilita a compreensão do funcionamento global do sistema e permite a sua adaptação para utilização em ambiente real.

De forma geral, o sistema pode ser dividido nos seguintes componentes principais:

1. **Aquisição da Imagem** – receção de imagens provenientes do conjunto de dados ou fornecidas pelo utilizador;
2. **Pré-processamento** – redimensionamento e normalização das imagens de acordo com os requisitos do modelo;
3. **Modelo de Classificação** – rede neuronal convolucional baseada em aprendizagem por transferência;
4. **Módulo de Otimização** – ajuste dos hiperparâmetros das camadas finais através de algoritmos swarm;
5. **Avaliação e Resultados** – geração de métricas de desempenho e análise dos resultados obtidos.

O fluxo de dados ao longo do sistema é sequencial, iniciando-se na entrada da imagem e culminando na previsão da classe correspondente.

3.2 Arquitetura do Modelo de Classificação

O modelo de classificação adotado baseia-se na arquitetura **MobileNetV2**, utilizada como rede base pré-treinada. Esta rede é responsável pela extração de características visuais de alto nível a partir das imagens de entrada.

Sobre a rede base foram adicionadas camadas finais responsáveis pela adaptação ao problema em estudo, constituídas por:

- uma camada de **Global Average Pooling**, para redução da dimensionalidade;
- uma camada **totalmente ligada** com número de neurónios ajustável;
- uma camada de **Dropout**, com valor fixo, para regularização;
- uma camada de saída **Softmax**, correspondente às cinco classes do problema.

A rede base permanece congelada durante o treino, sendo apenas atualizados os pesos das camadas adicionadas. Esta decisão reduz significativamente o custo computacional e melhora a capacidade de generalização do modelo.

3.3 Integração do Módulo de Otimização

O módulo de otimização integra os algoritmos PSO e WSA, responsáveis por ajustar os hiperparâmetros das camadas densas finais do modelo. Estes algoritmos interagem diretamente com o modelo de classificação através de uma função de fitness, que avalia o desempenho de cada configuração no conjunto de validação.

Esta separação entre o modelo de classificação e o módulo de otimização permite uma integração flexível, possibilitando a substituição ou extensão dos algoritmos de otimização sem alterar a arquitetura principal do sistema.

3.4 Preparação para Aplicação Web

Após a fase de treino e avaliação, o modelo final é guardado em ficheiro, permitindo a sua posterior reutilização. Este modelo pode ser carregado por uma aplicação web, destinada à classificação de novas imagens fornecidas pelo utilizador.

A aplicação web segue uma arquitetura simples, composta por:

- uma interface de utilizador para submissão de imagens;
- um módulo de pré-processamento idêntico ao utilizado durante o treino;
- o modelo treinado para geração da previsão;
- a apresentação do resultado ao utilizador.

Esta abordagem permite validar o desempenho do classificador em ambiente real e demonstra a aplicabilidade prática do sistema desenvolvido.

3.5 Arquitetura da Aplicação Web

Para demonstrar o sistema em funcionamento real e garantir a persistência do modelo treinado, foi desenvolvida uma aplicação web recorrendo à plataforma Streamlit. Esta aplicação integra o modelo final previamente treinado e guardado em ficheiro, permitindo a sua reutilização sem necessidade de novo treino.

A arquitetura da aplicação web é composta pelos seguintes componentes principais:

- Interface de Utilizador: responsável pela interação com o utilizador, permitindo a aquisição de imagens através do carregamento de ficheiros ou recorrendo à câmara do dispositivo;
- Módulo de Pré-processamento: responsável pelo redimensionamento e normalização das imagens de entrada, garantindo a compatibilidade com os requisitos do modelo treinado;

- Modelo Persistido: modelo de classificação baseado em aprendizagem por transferência, carregado a partir de ficheiro e utilizado exclusivamente para inferência;
- Módulo de Inferência: responsável pela geração da previsão da classe e do respetivo nível de confiança;
- Apresentação de Resultados: visualização da classe prevista e das probabilidades associadas, permitindo ao utilizador avaliar o comportamento do classificador.

O fluxo de funcionamento da aplicação inicia-se com a aquisição da imagem, seguindo-se o pré-processamento, a inferência pelo modelo treinado e, por fim, a apresentação do resultado ao utilizador. Este processo ocorre em tempo real, demonstrando a aplicabilidade prática do sistema desenvolvido, conforme solicitado no enunciado.

4. Descrição da Implementação dos Algoritmos

4.1 Estrutura Geral da Implementação

O código foi estruturado de forma modular, permitindo separar claramente as diferentes fases do processo, nomeadamente: preparação do conjunto de dados, definição do modelo, otimização de hiperparâmetros, treino final, avaliação e análise de sensibilidade. Esta organização facilita a compreensão do funcionamento do sistema e garante a correspondência direta entre cada etapa implementada e os requisitos definidos no enunciado.

4.2 Implementação do Modelo por Transferência de Aprendizagem

O modelo de classificação foi implementado com base na arquitetura MobileNetV2, pré-treinada no conjunto de dados ImageNet. Esta arquitetura foi carregada sem a camada de classificação original (`include_top=False`), sendo utilizada exclusivamente como extratora de características.

Os pesos da rede base foram mantidos congelados durante todo o processo de treino, garantindo que apenas as camadas adicionadas fossem ajustadas. Sobre esta base foram implementadas as seguintes camadas finais:

- uma camada de Global Average Pooling, responsável pela redução da dimensionalidade;
- uma camada totalmente ligada com ativação ReLU, cujo número de neurónios constitui um hiperparâmetro a otimizar;
- uma camada de Dropout, utilizada para regularização;
- uma camada de saída com ativação Softmax, correspondente ao número de classes do problema.

O modelo foi compilado utilizando o otimizador Adam e a função de perda sparse categorical cross-entropy.

4.3 Implementação da Função de Fitness

A função de fitness foi implementada de forma a avaliar automaticamente o desempenho de diferentes configurações de hiperparâmetros. Esta função recebe como entrada um vetor contendo o número de neurónios da camada densa e a taxa de aprendizagem do otimizador.

Para cada configuração, o modelo é criado e treinado durante um número reduzido de épocas utilizando subconjuntos fixos dos dados de treino e validação, definidos previamente à otimização. Esta estratégia permite reduzir significativamente o custo computacional associado à otimização. O desempenho é avaliado através da accuracy

no conjunto de validação, sendo devolvido o seu valor negativo, de modo a adequar o problema a uma formulação de minimização.

Para garantir estabilidade na avaliação do desempenho durante a otimização, foram utilizados subconjuntos fixos dos dados de treino e validação, definidos previamente à execução dos algoritmos de otimização.

Esta abordagem reduz o ruído na função de fitness, assegurando que diferenças observadas entre configurações de hiperparâmetros resultam efetivamente das escolhas realizadas e não de variações nos dados utilizados.

4.4 Implementação dos Algoritmos de Otimização Swarm

A otimização dos hiperparâmetros foi realizada através de dois algoritmos swarm distintos: Particle Swarm Optimization (PSO) e Whale Swarm Algorithm (WSA). Ambos os algoritmos foram utilizados para explorar o espaço de soluções definido pelos limites do número de neurónios da camada densa e da taxa de aprendizagem.

Cada algoritmo avalia sucessivas populações de soluções candidatas, recorrendo à função de fitness para medir o desempenho de cada configuração. No final do processo, é identificada a melhor solução global encontrada por cada algoritmo, correspondente à configuração de hiperparâmetros com melhor desempenho no conjunto de validação.

4.5 Implementação do Random Search

Para efeitos de comparação com os algoritmos swarm, foi também implementado um método de Random Search para a otimização dos hiperparâmetros.

O Random Search consiste na geração aleatória de configurações de hiperparâmetros dentro dos limites definidos, avaliando cada configuração através da mesma função de fitness utilizada pelos algoritmos swarm.

Foram realizadas várias iterações independentes, sendo registado o desempenho de cada configuração no conjunto de validação. No final, foi selecionada a configuração com melhor accuracy de validação.

Esta abordagem permite avaliar se os ganhos obtidos pelos algoritmos swarm resultam de uma exploração mais eficiente do espaço de soluções, quando comparados com uma estratégia puramente aleatória.

4.6 Seleção do Modelo Final

Após a execução dos algoritmos PSO, WSA e do método de Random Search, as melhores configurações de hiperparâmetros obtidas por cada abordagem foram comparadas com base na accuracy de validação.

O algoritmo que apresentou o melhor desempenho foi selecionado automaticamente, sendo os respetivos hiperparâmetros utilizados para a construção do modelo final.

Com esta configuração, o modelo foi novamente criado e treinado utilizando todo o conjunto de treino e um número superior de épocas, de forma a obter o classificador definitivo. Este modelo foi posteriormente guardado para avaliação no conjunto de teste independente e para utilização futura em ambiente real.

4.7 Avaliação no Conjunto de Teste

A avaliação final do modelo foi realizada recorrendo a um conjunto de teste independente, que não foi utilizado em nenhuma das fases anteriores. As previsões do modelo foram comparadas com as classes reais, sendo gerada a matriz de confusão correspondente, permitindo analisar detalhadamente o desempenho do classificador em cada classe.

4.8 Implementação da Análise de Sensibilidade ao Tamanho do Conjunto de Treino

Para avaliar o impacto do número de instâncias no desempenho do modelo, foi implementada uma análise de sensibilidade baseada na variação da percentagem de dados de treino utilizada. O modelo final foi treinado utilizando diferentes frações do conjunto de treino e avaliado no mesmo conjunto de teste independente.

Os resultados obtidos foram registados em ficheiro Excel, permitindo a posterior visualização e análise da evolução do desempenho em função do tamanho do conjunto de treino.

5. Análise de Resultados

5.1 Resultados da Otimização dos Hiperparâmetros

Nesta fase do trabalho, procedeu-se à otimização dos hiperparâmetros da rede densa das camadas finais do modelo, recorrendo a três abordagens distintas: Particle Swarm Optimization (PSO), Whale Swarm Algorithm (WSA) e Random Search.

Os resultados obtidos foram registados em ficheiro Excel, permitindo uma análise comparativa direta entre os métodos. Observa-se que o algoritmo WSA apresentou o melhor desempenho em termos de accuracy de validação, superando tanto o PSO como o Random Search.

O método de Random Search apresentou resultados intermédios, confirmando que uma exploração aleatória do espaço de soluções pode conduzir a configurações razoáveis, embora menos consistentes quando comparadas com algoritmos baseados em inteligência coletiva.

O gráfico apresentado ilustra esta comparação, permitindo visualizar de forma clara as diferenças de desempenho entre os métodos de otimização considerados.

Algoritmo	Neuronios	LearningRate	Val_Accuracy
PSO	512	0,00029696	0,519999981
WSA	283	0,001204167	0,680000007
Random Search	159	0,002743865	0,539999974

5.2 Sensibilidade do Modelo à Variação dos Hiperparâmetros

A análise de sensibilidade à variação dos hiperparâmetros permitiu avaliar o impacto do método de otimização no desempenho do modelo. Os resultados demonstram que diferentes configurações dos hiperparâmetros conduzem a variações significativas na accuracy de validação, sendo o WSA o método que apresentou a melhor solução encontrada.

Este comportamento confirma a importância da otimização dos hiperparâmetros da rede densa, mesmo quando se utiliza aprendizagem por transferência, uma vez que as camadas finais têm um papel determinante na adaptação do modelo ao problema em estudo.

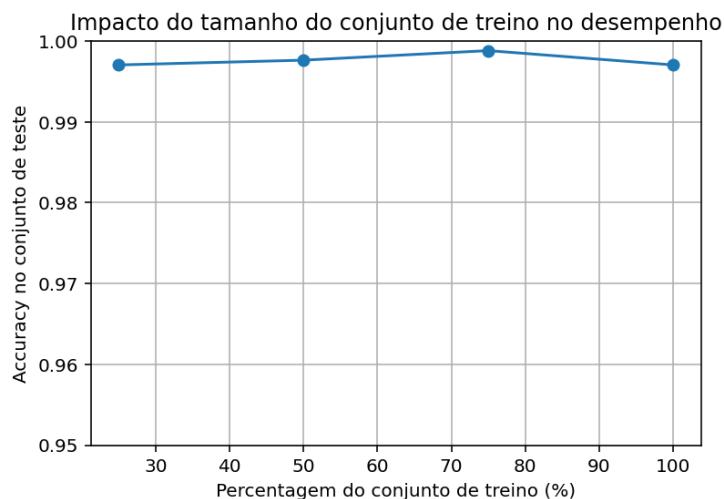
A comparação com o Random Search reforça a vantagem dos algoritmos swarm, uma vez que estes demonstram uma maior capacidade de encontrar soluções com melhor desempenho de forma mais consistente.

5.3 Impacto do Tamanho do Conjunto de Treino

Foi avaliado o impacto do número de instâncias de treino no desempenho do modelo. Para este efeito, o modelo final foi treinado utilizando diferentes percentagens do conjunto de treino, mantendo o conjunto de teste independente constante.

Os resultados obtidos mostram que o modelo alcança valores de accuracy elevados mesmo com uma percentagem reduzida de dados de treino, confirmando uma das principais vantagens da aprendizagem por transferência. Observa-se que, a partir de aproximadamente 50% do conjunto de treino, o desempenho do modelo se mantém estável, com variações pouco significativas na accuracy do conjunto de teste.

O gráfico apresentado evidencia esta tendência, demonstrando que a utilização de um número reduzido de exemplos de treino é suficiente para obter resultados satisfatórios neste problema.



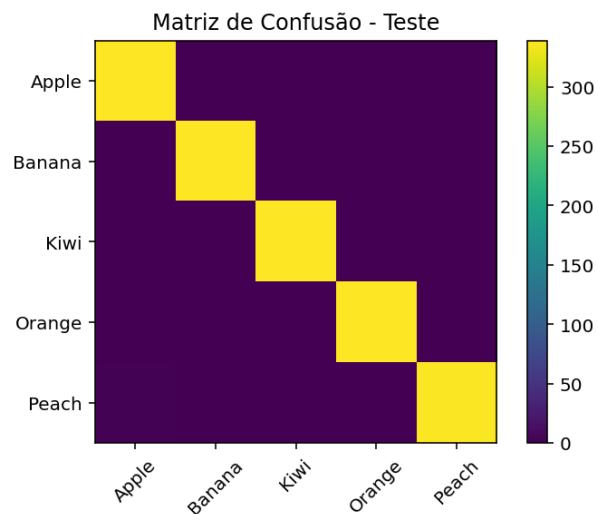
5.4 Avaliação Final no Conjunto de Teste Independente

A avaliação final do modelo foi realizada recorrendo a um conjunto de teste independente, não utilizado em nenhuma fase do treino ou validação. O desempenho obtido foi analisado através da matriz de confusão e das métricas de classificação, incluindo precisão, recall e F1-score.

Os resultados demonstram um desempenho extremamente elevado em todas as classes, com valores próximos de 100% para todas as métricas consideradas. A matriz de confusão confirma que o modelo classifica corretamente praticamente todas as amostras do conjunto de teste, não se observando confusões significativas entre classes.

Este desempenho evidencia a eficácia do modelo treinado por transferência de aprendizagem, bem como a adequação da arquitetura escolhida e do processo de otimização dos hiperparâmetros.

	precision	recall	f1-score	support
Apple	0.99	1.00	1.00	339
Banana	1.00	1.00	1.00	339
Kiwi	1.00	1.00	1.00	338
Orange	1.00	1.00	1.00	339
Peach	1.00	0.99	1.00	339
accuracy			1.00	1694
macro avg	1.00	1.00	1.00	1694
weighted avg	1.00	1.00	1.00	1694



5.5 Discussão dos Resultados

De forma global, os resultados obtidos demonstram que a utilização de aprendizagem por transferência, aliada à otimização das camadas densas finais, permite alcançar um elevado desempenho mesmo com um conjunto de dados relativamente reduzido. A comparação entre os algoritmos PSO e WSA evidenciou vantagens do WSA neste contexto, embora ambos tenham permitido obter soluções válidas.

A análise do impacto do tamanho do conjunto de treino confirma que a aprendizagem por transferência reduz significativamente a necessidade de grandes volumes de dados, cumprindo assim os objetivos definidos no enunciado.

6. Conclusões

Neste trabalho foi desenvolvido um sistema de classificação de imagens recorrendo a aprendizagem por transferência, com base numa rede neuronal convolucional pré-treinada. A utilização da arquitetura MobileNetV2 permitiu aproveitar representações visuais previamente aprendidas, reduzindo significativamente o custo computacional do treino e melhorando o desempenho do modelo, mesmo com um conjunto de dados de dimensão moderada.

A otimização dos hiperparâmetros das camadas densas finais revelou-se um passo importante para a adaptação do modelo ao problema em estudo. A comparação entre os algoritmos de otimização PSO e WSA demonstrou que ambos são adequados para este tipo de tarefa, tendo o WSA apresentado melhores resultados em termos de accuracy de validação. Esta análise confirmou a relevância da otimização de hiperparâmetros, mesmo em cenários de aprendizagem por transferência.

A inclusão do método de Random Search permitiu validar experimentalmente a eficácia dos algoritmos swarm, demonstrando que estes apresentam uma exploração mais eficiente do espaço de soluções quando comparados com abordagens aleatórias.

A avaliação do impacto do tamanho do conjunto de treino evidenciou uma das principais vantagens da aprendizagem por transferência, nomeadamente a capacidade de obter resultados elevados com um número reduzido de exemplos de treino. Verificou-se que o desempenho do modelo se mantém estável a partir de uma fração relativamente pequena dos dados disponíveis, o que é particularmente relevante em aplicações reais onde a recolha e anotação de dados pode ser limitada.

A avaliação final no conjunto de teste independente revelou um desempenho muito elevado, com valores próximos de 100% para as métricas de precisão, recall e F1-score em todas as classes. A matriz de confusão confirmou a capacidade do modelo em distinguir corretamente as diferentes classes, não se observando confusões significativas.

Por fim, o modelo treinado foi integrado numa aplicação web, permitindo demonstrar o sistema em funcionamento real, com classificação de novas imagens em tempo real. Esta aplicação valida a persistência do modelo e evidencia a aplicabilidade prática da solução desenvolvida, cumprindo integralmente os objetivos definidos no enunciado.

Em suma, os resultados obtidos demonstram que a combinação de aprendizagem por transferência com técnicas de otimização baseadas em inteligência coletiva constitui uma abordagem eficaz para problemas de classificação de imagens, apresentando bom desempenho, eficiência computacional e aplicabilidade em contextos reais.

7. Bibliografia e Referências

<https://streamlit.io/cloud>

<https://chatgpt.com>

<https://keras.io/api/applications/mobilenet/>

<https://chatgpt.com/share/693f8595-bdb0-800d-920b-5c9873d574e5>

<https://chatgpt.com/share/693f85b9-2298-800d-af48-ddbec46941e0>