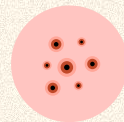
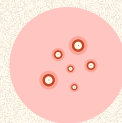




DERMNET

Classificação de Doenças de Pele





GRUPO 6



Daniel Valadares de Souza Felix
Gustavo Silvestre Almeida Conceição
João Vitor Lima de Melo
Larissa Valadares Silqueira
Leonardo Barbosa Brandão





SUMÁRIO



01

Introdução

Contextualização e objetivos

02

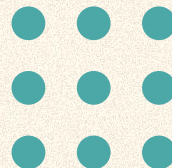
Desenvolvimento

Front-end;
Rede Neural;
Computação Paralela;
Back-end;
Computação Distribuída

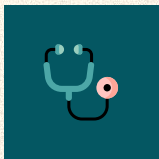
03

Conclusão

Considerações e propostas



INTRODUÇÃO



Contextualização

- Aplicação de técnicas de processamento de imagens e aprendizado de máquina na medicina.
- Detecção precoce de doenças de pele para tratamento eficaz.

Objetivos

- Fornecer suporte confiável para diagnósticos preliminares no ambiente acadêmico e hospitalar.
- Desenvolver um sistema capaz de classificar diversas doenças de pele a partir de imagens.

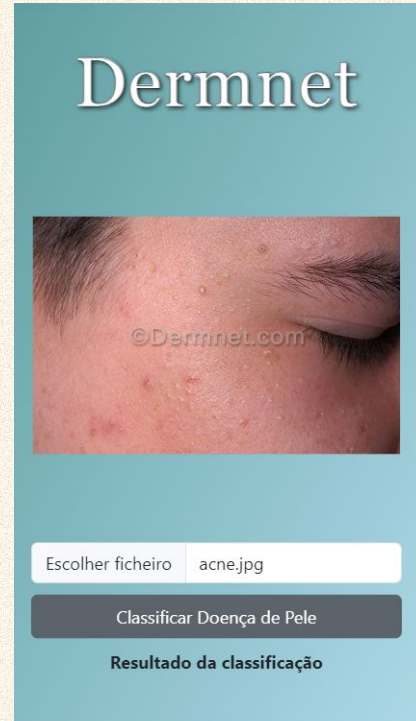


DESENVOLVIMENTO



Front-end

- Desenvolvimento realizado no VSCode
- Utilização de HTML, CSS e JavaScript para construir a interface do usuário.
- Implementação de um design responsivo.



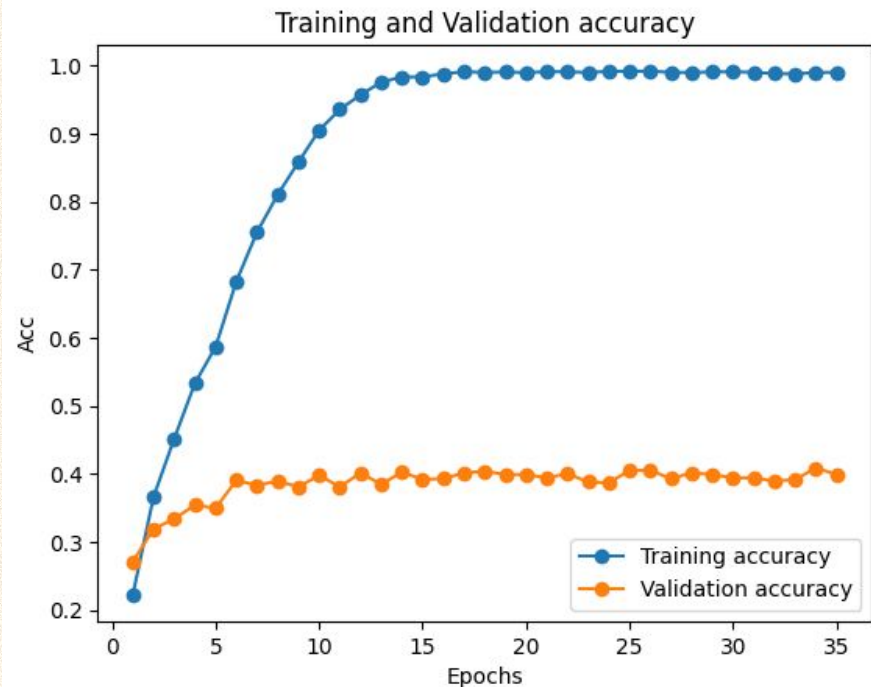
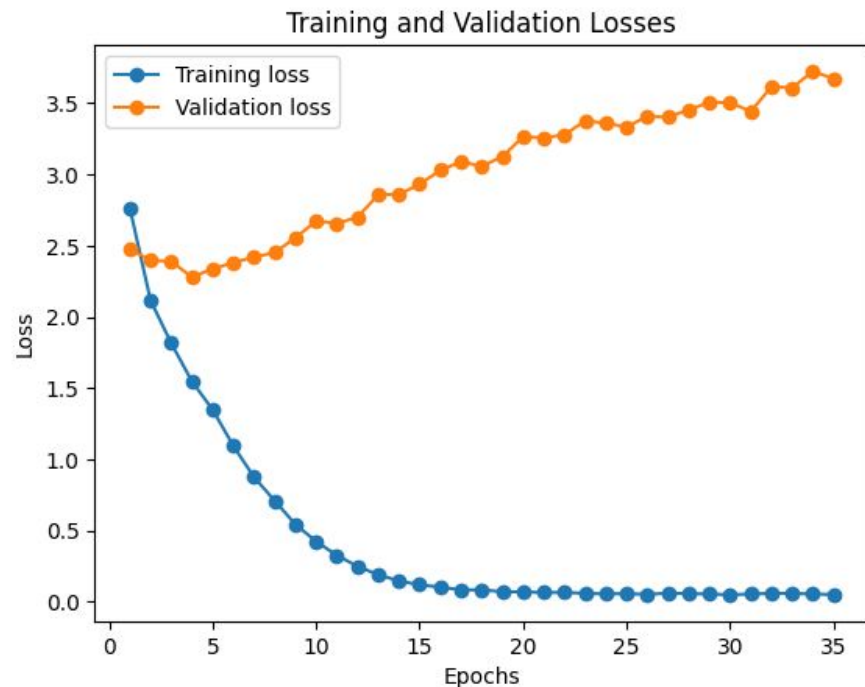
DESENVOLVIMENTO

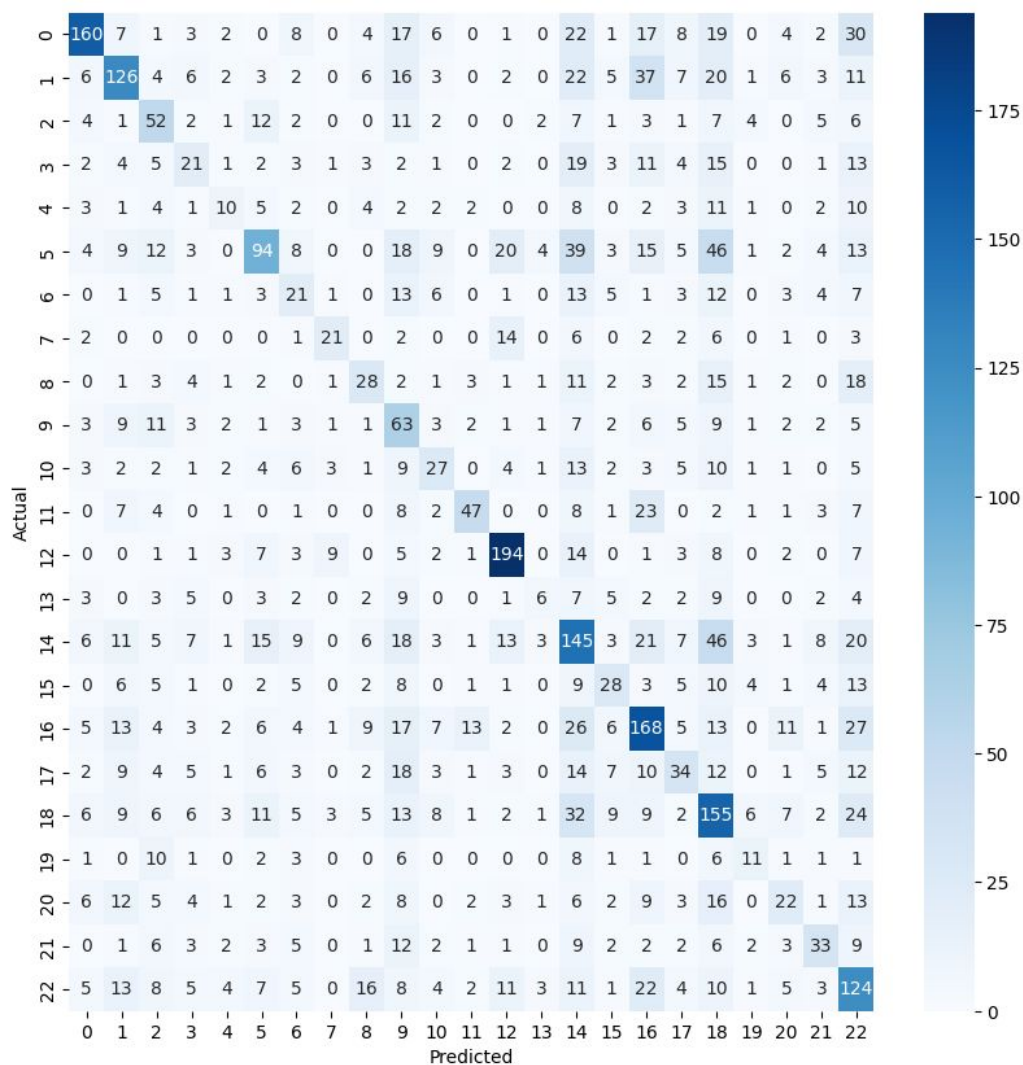
Rede Neural



- Utilização do Google Colab para treinamento e teste da base
- Implementação da rede neural ResNet50
- Execução em GPU com 2560 CUDA cores (Tesla T4)
- 4.608 imagens para o treinamento
- Tempo de Treinamento: 637 segundos ~ 18,2 seg/época
- Média da Acurácia no Treino: 88%
- Média da Acurácia na Validação: 38%

DESENVOLVIMENTO





DESENVOLVIMENTO

Rede Neural

- Teste com 4.000 imagens
- Matriz de Confusão
- Acurácia de ~40%



DESENVOLVIMENTO

Computação Paralela



- Treinamento com 5 classes arbitrárias
- Intel Xeon CPU 2.20 GHz :: 2 cores
- GPU NVIDIA Tesla T4 :: 2560 CUDA cores
- Arquiteturas:
 - Sequencial em CPU
 - Paralelo em CPU
 - GPU





DESENVOLVIMENTO



Informações Obtidas

Arquitetura	Média de Tempo por Época	Tempo de Execução Total	Acurácia Média no Treino	Acurácia Média na Validação
Sequencial	207 seg	2491 seg	87%	69%
Paralelo	177 seg	2127 seg	88%	70%
GPU	7 seg	86 seg	89%	71%





DESENVOLVIMENTO



Avaliação de Desempenho

Arquitetura	Tempo de Execução	Ganho	Eficiência
Sequencial em CPU	2491 seg		
Paralelo em CPU	2127 seg	1,2	0,6
GPU	86 seg	29	0,01



DESENVOLVIMENTO

Back-end



- Utilização do framework Flask
- O usuário envia uma imagem através da interface e o Flask processa a requisição
- O classificador retorna a classe à qual a imagem pertence, exibindo a resposta para o usuário

DESENVOLVIMENTO



Computação Distribuída – Detalhes

- Distribuição via Web Service
- Capacidade de escalar recursos de computação e armazenamento conforme a demanda do sistema
- Configuração do ambiente para garantir o funcionamento adequado do sistema

DESENVOLVIMENTO

Computação Distribuída – Serviço de Nuvem

- Hospedagem pela Amazon Web Services (AWS)
- Uso do serviço da Amazon Elastic Compute Cloud (Amazon EC2)
- Instância do tipo g4dn.xlarge
 - 1 GPUs NVIDIA T4 e tecnologia RTX
 - CPU Intel Cascade Lake (c/ 4 vCPUs)
 - 16 GB de Memória
 - Armazenamento de 1 x 125 SSD NVMe



CONCLUSÃO



A implementação parcial do projeto foi bem sucedida. Para a próxima etapa, serão realizados os ajustes necessários com base nos resultados obtidos até o momento, visando aprimorar ainda mais a precisão e eficiência do sistema.

Além disso, será feita a integração completa do sistema, conectando de forma eficaz o front-end com o back-end e disponibilizando o sistema para uso público, a partir da hospedagem em nuvem.





FIM

