

# Aplicação de deep learning em dispositivos Android

Tiago de Miranda Leite

NUSP: 7595289

[tiago.miranda.leite@usp.br](mailto:tiago.miranda.leite@usp.br)

Orientador: Prof. Dr. João do Espírito Santo Batista Neto

Apresentação do Trabalho de Conclusão de Curso

Bacharelado em Ciências de Computação  
Instituto de Ciências Matemáticas e de Computação – ICMC  
Universidade de São Paulo - USP

28 de Novembro de 2018

# Sumário

Introdução

Métodos

Desenvolvimento

Conclusão

Referências

# Sumário

Introdução

Métodos

Desenvolvimento

Conclusão

Referências

# Introdução

## Contextualização e motivação

- Modelos *deep learning* têm sido aplicados em diversos problemas atualmente:
  - Classificação de imagens
  - Séries temporais
  - Processamento de linguagem natural
- Em classificação de imagens, redes convolucionais são os modelos mais bem-sucedidos.
- Crescente surgimento de aplicativos inteligentes para smartphones:
  - Filtros artísticos
  - Detecção de rostos
- Existência de modelos de redes convolucionais otimizados para dispositivos móveis, como a redMobileNet [?]

# Introdução

## Contextualização e motivação

- Modelos *deep learning* têm sido aplicados em diversos problemas atualmente:
  - Classificação de imagens
  - Séries temporais
  - Processamento de linguagem natural
- Em classificação de imagens, redes convolucionais são os modelos mais bem-sucedidos.
- Crescente surgimento de aplicativos inteligentes para smartphones:
  - Filtros artísticos
  - Detecção de rostos
- Existência de modelos de redes convolucionais otimizados para dispositivos móveis, como a redMobileNet [?]

# Introdução

## Contextualização e motivação

- Modelos *deep learning* têm sido aplicados em diversos problemas atualmente:
  - Classificação de imagens
  - Séries temporais
  - Processamento de linguagem natural
- Em classificação de imagens, redes convolucionais são os modelos mais bem-sucedidos.
- Crescente surgimento de aplicativos inteligentes para smartphones:
  - Filtros artísticos
  - Detecção de rostos
- Existência de modelos de redes convolucionais otimizados para dispositivos móveis, como a redMobileNet [?]

# Introdução

## Contextualização e motivação

- Modelos *deep learning* têm sido aplicados em diversos problemas atualmente:
  - Classificação de imagens
  - Séries temporais
  - Processamento de linguagem natural
- Em classificação de imagens, redes convolucionais são os modelos mais bem-sucedidos.
- Crescente surgimento de aplicativos inteligentes para smartphones:
  - Filtros artísticos
  - Detecção de rostos
- Existência de modelos de redes convolucionais otimizados para dispositivos móveis, como a redMobileNet [?]

# Introdução

## Objetivos

Este trabalho teve como objetivo:

- ① Treinamento de 8 variações de rede MobileNet, utilizando transferência de conhecimento;
- ② Análise do desempenho de classificação das redes treinadas;
- ③ Criação de um aplicativo Android para classificação de espécies de flores em imagens, utilizando o melhor modelo obtido.

# Introdução

## Objetivos

Este trabalho teve como objetivo:

- ① Treinamento de 8 variações de rede MobileNet, utilizando transferência de conhecimento;
- ② Análise do desempenho de classificação das redes treinadas;
- ③ Criação de um aplicativo Android para classificação de espécies de flores em imagens, utilizando o melhor modelo obtido.

# Introdução

## Objetivos

Este trabalho teve como objetivo:

- ① Treinamento de 8 variações de rede MobileNet, utilizando transferência de conhecimento;
- ② Análise do desempenho de classificação das redes treinadas;
- ③ Criação de um aplicativo Android para classificação de espécies de flores em imagens, utilizando o melhor modelo obtido.

# Sumário

Introdução

Métodos

Desenvolvimento

Conclusão

Referências

# Métodos

## Linguagens e tecnologias utilizadas

- Treinamento das redes:
  - Python versão 3.5;
  - Biblioteca TensorFlow versão 1.10.0.
- Implementação do aplicativo:
  - Ambiente Android Studio (Java);
  - TensorFlow para Android versão 1.10.0.

# Métodos

## Linguagens e tecnologias utilizadas

- Treinamento das redes:
  - Python versão 3.5;
  - Biblioteca TensorFlow versão 1.10.0.
- Implementação do aplicativo:
  - Ambiente Android Studio (Java);
  - TensorFlow para Android versão 1.10.0.

# Métodos

## Métricas de avaliação das redes

### Avaliação das redes:

- Fase 1: Acurácia das redes no conjunto de validação, durante o treinamento;
- Fase 2: Acurácia, precisão, *recall* e *F-score* [?] das melhores redes da fase 1, no conjunto de teste, após o treinamento. O tamanho final da rede também foi considerado.

# Métodos

## Métricas de avaliação das redes

### Avaliação das redes:

- Fase 1: Acurácia das redes no conjunto de validação, durante o treinamento;
- Fase 2: Acurácia, precisão, *recall* e *F-score* [?] das melhores redes da fase 1, no conjunto de teste, após o treinamento. O tamanho final da rede também foi considerado.

# Métodos

## Métodos, Técnicas e Tecnologias Utilizadas

Métrica	Expressao	Descricao
Acurácia	$\sum_{i=1}^k \frac{tp_i + tn_i}{tp_i + fp_i + tn_i + fn_i}$	Lorem Ipsum
Precisão	$\sum_{n=1}^{\infty} 4^{-n}$	Lorem Ipsum

# Sumário

Introdução

Métodos

Desenvolvimento

Conclusão

Referências

# Desenvolvimento

## Descrição do Problema

- Obtenção de um conjunto de dados de imagens para treinamento, validação e teste
- Criação de variações da rede MobileNet.
- Realização do treinamento das redes criadas, utilizando transferência de conhecimento.
- Comparação entre as redes treinadas.
- Implementação do aplicativo e utilização do melhor modelo.

# Desenvolvimento

## Atividades realizadas

- Levantamento das espécies de flores mais comuns nas redondezas do câmpus;
- Obtenção do conjunto de dados:
  - 16 classes;
  - Google imagens e Oxford 102 Category Flowers dataset;
  - Total de 4713 imagens: 70% para treinamento, 15% para validação e 20% para teste.
- Obtenção da rede MobileNet:
  - Google AI Blog;
  - Modelos já treinados com o conjunto de dados Imagenet;
  - Tamanho da imagem de entrada: 224x224x3;
  - Tamanhos dos mapas de atributos das camadas de convolução: 25%, 50%, 75% e 100%.

# Desenvolvimento

## Atividades realizadas

- Levantamento das espécies de flores mais comuns nas redondezas do câmpus;
- Obtenção do conjunto de dados:
  - 16 classes;
  - Google imagens e Oxford 102 Category Flower Dataset.
  - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
  - Google AI Blog;
  - Modelos já treinados com o conjunto de dados ImageNet;
  - Tamanho da imagem de entrada: 224x224x3;
  - Tamanhos dos mapas de atributos das camadas de convolução: 25%, 50%, 75% e 100%.

# Desenvolvimento

## Atividades realizadas

- Levantamento das espécies de flores mais comuns nas redondezas do câmpus;
- Obtenção do conjunto de dados:
  - 16 classes;
  - Google imagens e Oxford 102 Category Flower Dataset.
  - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
  - Google AI Blog;
  - Modelos já treinados com o conjunto de dados ImageNet;
  - Tamanho da imagem de entrada: 224x224x3;
  - Tamanhos dos mapas de atributos das camadas de convolução: 25%, 50%, 75% e 100%.

# Desenvolvimento

## Atividades realizadas

- Levantamento das espécies de flores mais comuns nas redondezas do câmpus;
- Obtenção do conjunto de dados:
  - 16 classes;
  - Google imagens e Oxford 102 Category Flower Dataset.
  - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
  - Google AI Blog;
  - Modelos já treinados com o conjunto de dados ImageNet;
  - Tamanho da imagem de entrada: 224x224x3;
  - Tamanhos dos mapas de atributos das camadas de convolução: 25%, 50%, 75% e 100%.

# Desenvolvimento

## Atividades realizadas

- Levantamento das espécies de flores mais comuns nas redondezas do câmpus;
- Obtenção do conjunto de dados:
  - 16 classes;
  - Google imagens e Oxford 102 Category Flower Dataset.
  - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
  - Google AI Blog;
  - Modelos já treinados com o conjunto de dados ImageNet;
  - Tamanho da imagem de entrada: 224x224x3;
  - Tamanhos dos mapas de atributos das camadas de convolução: 25%, 50%, 75% e 100%.

# Desenvolvimento

## Atividades realizadas

- Levantamento das espécies de flores mais comuns nas redondezas do câmpus;
- Obtenção do conjunto de dados:
  - 16 classes;
  - Google imagens e Oxford 102 Category Flower Dataset.
  - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
  - Google AI Blog;
  - Modelos já treinados com o conjunto de dados ImageNet;
  - Tamanho da imagem de entrada: 224x224x3;
  - Tamanhos dos mapas de atributos das camadas de convolução: 25%, 50%, 75% e 100%.

# Desenvolvimento

## Atividades realizadas

- Levantamento das espécies de flores mais comuns nas redondezas do câmpus;
- Obtenção do conjunto de dados:
  - 16 classes;
  - Google imagens e Oxford 102 Category Flower Dataset.
  - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
  - Google AI Blog;
  - Modelos já treinados com o conjunto de dados ImageNet;
  - Tamanho da imagem de entrada: 224x224x3;
  - Tamanhos dos mapas de atributos das camadas de convolução: 25%, 50%, 75% e 100%.

# Desenvolvimento

## Atividades realizadas

- Levantamento das espécies de flores mais comuns nas redondezas do câmpus;
- Obtenção do conjunto de dados:
  - 16 classes;
  - Google imagens e Oxford 102 Category Flower Dataset.
  - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
  - Google AI Blog;
  - Modelos já treinados com o conjunto de dados ImageNet;
  - Tamanho da imagem de entrada: 224x224x3;
  - Tamanhos dos mapas de atributos das camadas de convolução: 25%, 50%, 75% e 100%.

# Desenvolvimento

## Atividades realizadas

- Alterações realizadas: Modelo 1

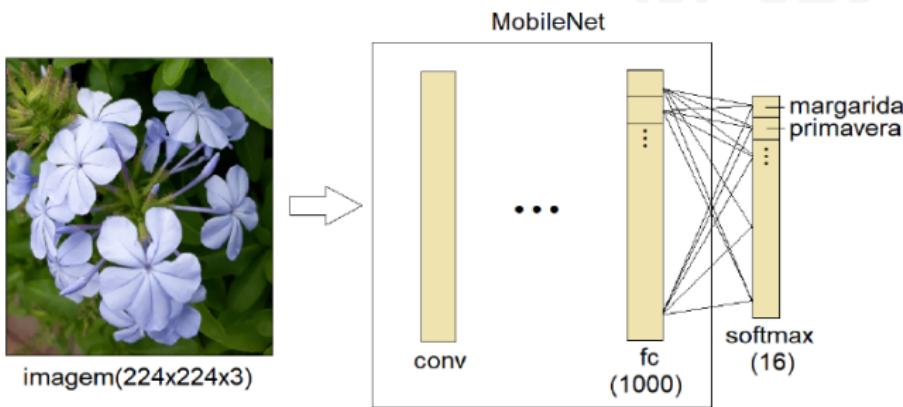


Figura: Modelo 1 [?]

# Desenvolvimento

Atividades realizadas

- Alterações realizadas: Modelo 2

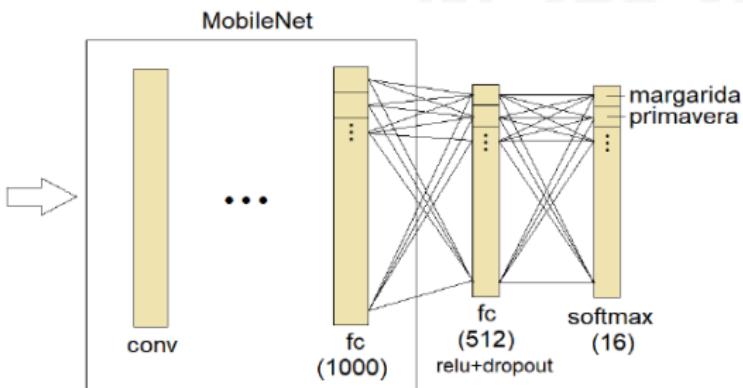


Figura: Modelo 2 [?]

# Desenvolvimento

## Atividades realizadas

- Resumo das alterações:

Grupo	Nome	Fração dos mapas de atributos	Camadas adicionadas	Dropout
1	MobileNet_1_25	25%	softmax (16)	Não
1	MobileNet_1_50	50%	softmax (16)	Não
1	MobileNet_1_75	75%	softmax (16)	Não
1	MobileNet_1_100	100%	softmax (16)	Não
2	MobileNet_2_25	25%	fc(512) e softmax (16)	Sim
2	MobileNet_2_50	50%	fc(512) e softmax (16)	Sim
2	MobileNet_2_75	75%	fc(512) e softmax (16)	Sim
2	MobileNet_2_100	100%	fc(512) e softmax (16)	Sim

Figura: Resumos [?]

# Desenvolvimento

## Atividades realizadas

- Treinamento dos modelos propostos:
  - Lotes de 100 imagens aleatórias;
  - 5000 iterações.
- Implementação do aplicativo.

# Desenvolvimento

## Atividades realizadas

- Treinamento dos modelos propostos:
  - Lotes de 100 imagens aleatórias;
  - 5000 iterações.
- Implementação do aplicativo.

# Desenvolvimento

## Atividades realizadas

- Treinamento dos modelos propostos:
  - Lotes de 100 imagens aleatórias;
  - 5000 iterações.
- Implementação do aplicativo.

# Desenvolvimento

## Atividades realizadas

- Treinamento dos modelos propostos:
  - Lotes de 100 imagens aleatórias;
  - 5000 iterações.
- Implementação do aplicativo.

# Desenvolvimento

## Resultados e Discussão

### Treinamentos e arquitetura escolhida

- Grupo 1

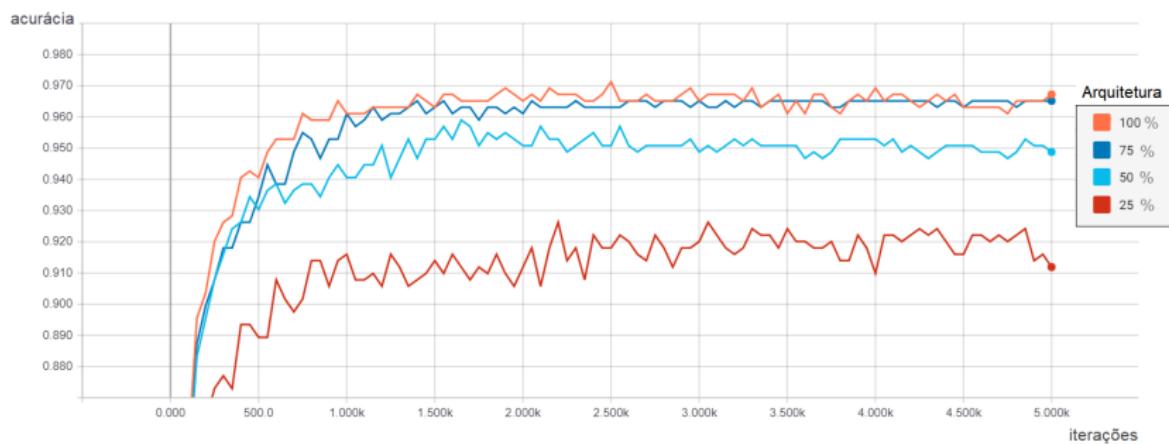


Figura: Tiago

# Desenvolvimento

## Resultados e Discussão

### Treinamentos e arquitetura escolhida

- Grupo 2

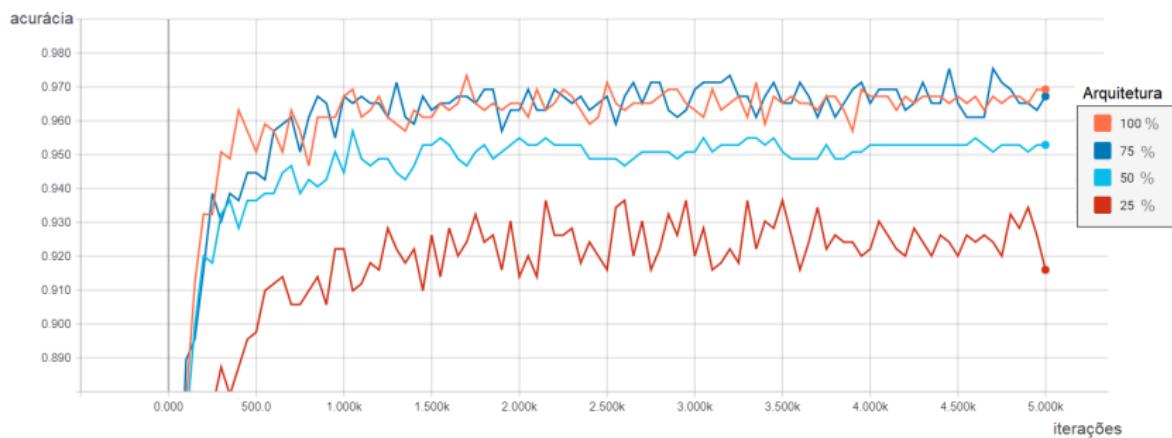


Figura: Tiago

# Desenvolvimento

## Resultados e Discussão

### Treinamentos e arquitetura escolhida

- Maiores valores de acurácia para:
  - Grupo 1: MobileNet\_1\_75 e MobileNet\_1\_100
  - Grupo 2: MobileNet\_2\_75 e MobileNet\_2\_100

# Desenvolvimento

## Resultados e Discussão

### Treinamentos e arquitetura escolhida

- Maiores valores de acurácia para:
  - Grupo 1: MobileNet\_1\_75 e MobileNet\_1\_100
  - Grupo 2: MobileNet\_2\_75 e MobileNet\_2\_100

Avaliação das duas melhores arquiteturas de cada grupo no conjunto de teste:

# Desenvolvimento

## Resultados e Discussão

### Treinamentos e arquitetura escolhida

- Maiores valores de acurácia para:
  - Grupo 1: MobileNet\_1\_75 e MobileNet\_1\_100
  - Grupo 2: MobileNet\_2\_75 e MobileNet\_2\_100

Avaliação das duas melhores arquiteturas de cada grupo no conjunto de teste:

Nome	Acurácia	Precisão	Recall	F-score	Tamanho
Mobilenet_1_100	0,954	0,955	0,953	0,954	17,2 MB
Mobilenet_2_100	0,961	0,961	0,959	0,960	19,2 MB
Mobilenet_1_75	0,955	0,956	0,956	0,956	10,6 MB
Mobilenet_2_75	0,955	0,957	0,956	0,956	12,6 MB

Figura: Resumos [?]

# Desenvolvimento

## Resultados e Discussão

### Aplicativo

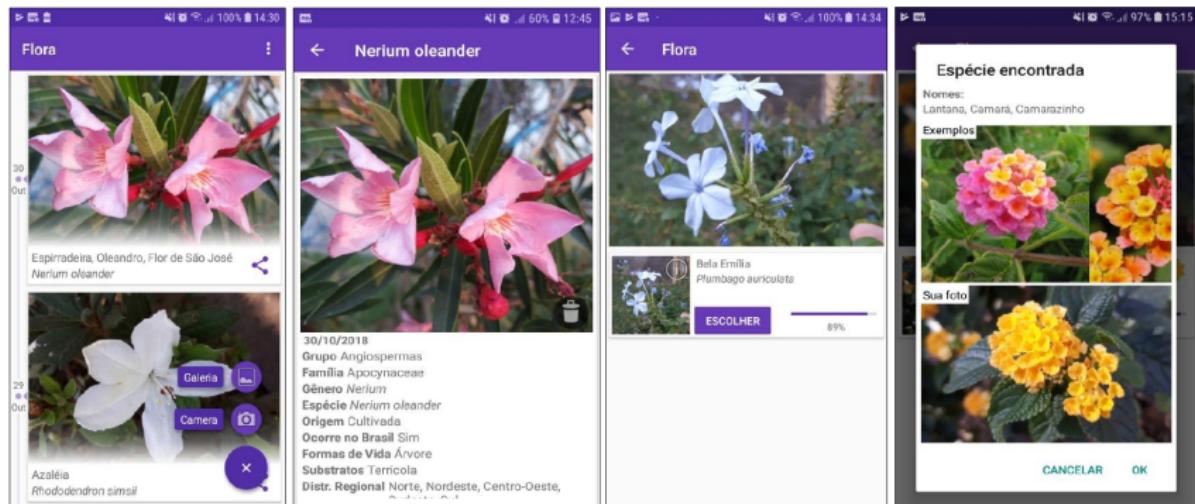


Figura: Resumos [?]

# Sumário

Introdução

Métodos

Desenvolvimento

Conclusão

Referências

# Conclusão

## Contribuições:

- Confirmação da possibilidade de reutilização, refinamento e manipulação de modelos de redes neurais convolucionais, pré-treinados, para classificação de imagens aplicadas a um contexto mais específico.
- Demonstração da utilização, em ambiente Android, do modelo treinado, através de um aplicativo que pode ser usado por qualquer pessoa.
- Colaboração da difusão de conhecimentos de botânica.

# Conclusão

## Contribuições:

- Confirmação da possibilidade de reutilização, refinamento e manipulação de modelos de redes neurais convolucionais, pré-treinados, para classificação de imagens aplicadas a um contexto mais específico.
- Demonstração da utilização, em ambiente Android, do modelo treinado, através de um aplicativo que pode ser usado por qualquer pessoa.
- Colaboração da difusão de conhecimentos de botânica.

# Conclusão

## Contribuições:

- Confirmação da possibilidade de reutilização, refinamento e manipulação de modelos de redes neurais convolucionais, pré-treinados, para classificação de imagens aplicadas a um contexto mais específico.
- Demonstração da utilização, em ambiente Android, do modelo treinado, através de um aplicativo que pode ser usado por qualquer pessoa.
- Colaboração da difusão de conhecimentos de botânica.

# Conclusão

## Dificuldades e Limitações:

- Indisponibilidade de hardware (GPU) para treinamento mais rápido das redes, o que inviabilizou a experimentação de diversos parâmetros;
- Ocorrência de erros na execução da rede no aplicativo devido à mudança de linguagem (Python → Java);
- Conjunto de dados limitado.

# Conclusão

## Dificuldades e Limitações:

- Indisponibilidade de hardware (GPU) para treinamento mais rápido das redes, o que inviabilizou a experimentação de diversos parâmetros;
- Ocorrência de erros na execução da rede no aplicativo devido à mudança de linguagem (Python → Java);
- Conjunto de dados limitado.

# Conclusão

## Dificuldades e Limitações:

- Indisponibilidade de hardware (GPU) para treinamento mais rápido das redes, o que inviabilizou a experimentação de diversos parâmetros;
- Ocorrência de erros na execução da rede no aplicativo devido à mudança de linguagem (Python → Java);
- Conjunto de dados limitado.

# Conclusão

## Dificuldades e Limitações:

- Indisponibilidade de hardware (GPU) para treinamento mais rápido das redes, o que inviabilizou a experimentação de diversos parâmetros;
- Ocorrência de erros na execução da rede no aplicativo devido à mudança de linguagem (Python → Java);
- Conjunto de dados limitado.

## Trabalhos futuros:

- Treinamento de camadas de convolução e aumento do conjunto de dados e do número de classes;
- Colaboração do usuário na expansão do conjunto de dados.

# Conclusão

## Dificuldades e Limitações:

- Indisponibilidade de hardware (GPU) para treinamento mais rápido das redes, o que inviabilizou a experimentação de diversos parâmetros;
- Ocorrência de erros na execução da rede no aplicativo devido à mudança de linguagem (Python → Java);
- Conjunto de dados limitado.

## Trabalhos futuros:

- Treinamento de camadas de convolução e aumento do conjunto de dados e do número de classes;
- Colaboração do usuário na expansão do conjunto de dados.

# Sumário

Introdução

Métodos

Desenvolvimento

Conclusão

Referências

# Referência Bibliográfica I



Obrigado!

Dúvidas?

## Aplicação de deep learning em dispositivos Android

Tiago de Miranda Leite

NUSP: 7595289

[tiago.miranda.leite@usp.br](mailto:tiago.miranda.leite@usp.br)

Orientador: Prof. Dr. João do Espírito Santo Batista Neto

Apresentação do Trabalho de Conclusão de Curso

Bacharelado em Ciências de Computação

Instituto de Ciências Matemáticas e de Computação – ICMC  
Universidade de São Paulo - USP