

Aplicação de deep learning em dispositivos Android

Tiago de Miranda Leite

NUSP: 7595289

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 rede MobileNet [3]

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 rede MobileNet [3]

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 rede MobileNet [3]

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 rede MobileNet [3]

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.
 - * Processamento local, sem requerer conexão à Internet.

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.
 - * Processamento local, sem requerer conexão à Internet.

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.
 - Processamento local, sem requerer conexão à Internet.

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* [5] 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* [5] das melhores redes da fase 1, no conjunto de teste, após o treinamento. O tamanho final da rede também foi considerado.

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 Flower Dataset [4].
 - Total de 4713 imagens: 70% para treinamento, 15% para validação e 20% para teste.
- Obtenção da rede MobileNet:
 - Google AI Blog [2];
 - Modelos já treinados com o conjunto de dados ImageNet [1].
 - 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 [4].
 - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
 - Google AI Blog [2];
 - Modelos já treinados com o conjunto de dados Imagenet-1000;
 - 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 [4].
 - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
 - Google AI Blog [2];
 - Modelos já treinados com o conjunto de dados Imagenet-1000;
 - 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 [4].
 - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
 - Google AI Blog [2];
 - Modelos já treinados com o conjunto de dados Imagens [4];
 - 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 [4].
 - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
 - Google AI Blog [2];
 - Modelos já treinados com o conjunto de dados Imagenet [3];
 - 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 [4].
 - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
 - Google AI Blog [2];
 - Modelos já treinados com o conjunto de dados ImageNet [1];
 - 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 [4].
 - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
 - Google AI Blog [2];
 - Modelos já treinados com o conjunto de dados ImageNet [1];
 - 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 [4].
 - Total de 4713 imagens: 70% para treinamento, 10% para validação e 20% para teste.
- Obtenção da rede MobileNet:
 - Google AI Blog [2];
 - Modelos já treinados com o conjunto de dados ImageNet [1];
 - 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

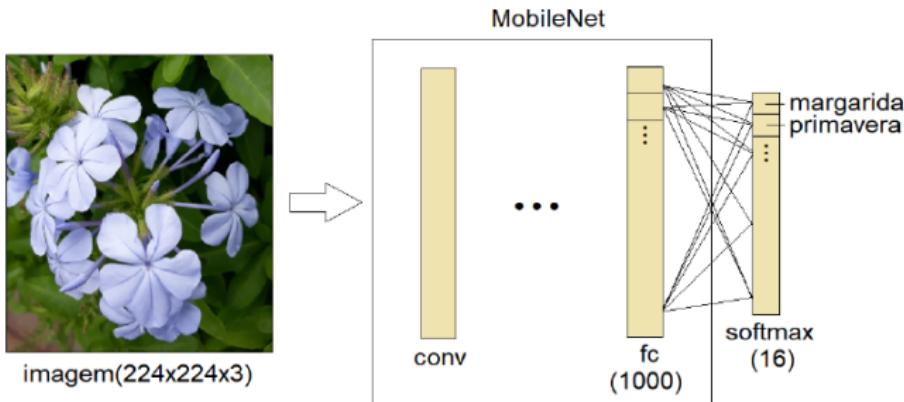


Figura: Esquema do primeiro modelo proposto. Fonte: elaborada pelo autor.

Desenvolvimento

Atividades realizadas

- Alterações realizadas: Modelo 2

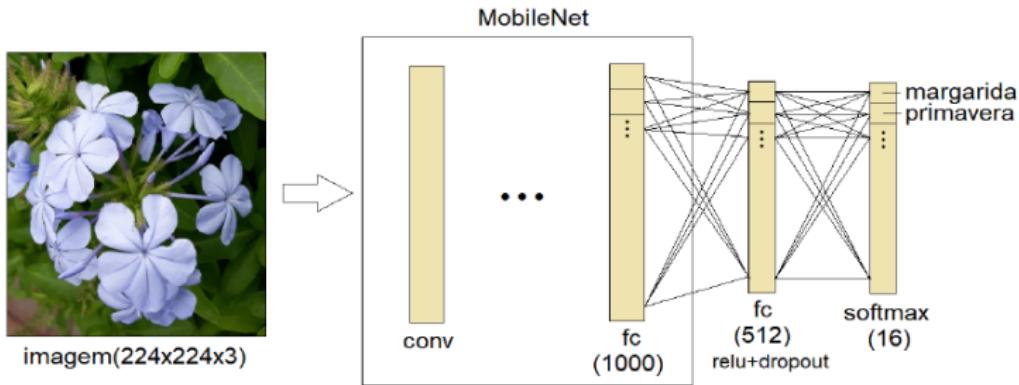


Figura: Esquema do segundo modelo proposto. Fonte: elaborada pelo autor.

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: Modelos de rede testados e suas características. Fonte: elaborada pelo autor.

Desenvolvimento

Atividades realizadas

- Treinamento dos modelos propostos:
 - Lotes de 100 imagens aleatórias;
 - 5000 iterações;
 - Acompanhamento em tempo real pela TensorBoard.
- Implementação do aplicativo.
 - Android Studio;
 - Versionamento com Git;
 - Banco de dados orientado a objetos Realm.

Desenvolvimento

Atividades realizadas

- Treinamento dos modelos propostos:
 - Lotes de 100 imagens aleatórias;
 - 5000 iterações;
 - Acompanhamento em tempo real pela TensorBoard.
- Implementação do aplicativo.
 - Android Studio;
 - Versionamento com Git;
 - Banco de dados orientado a objetos Realm.

Desenvolvimento

Resultados e Discussão

Treinamentos e arquitetura escolhida

- Grupo 1

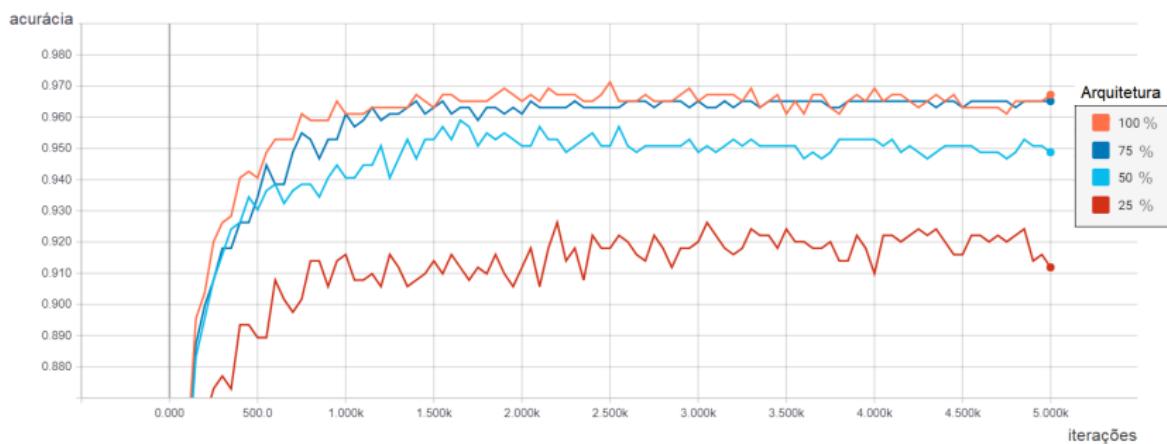


Figura: Acurácia das redes do grupo 1, no conjunto de validação, ao longo do treinamento.

Desenvolvimento

Resultados e Discussão

Treinamentos e arquitetura escolhida

- Grupo 2

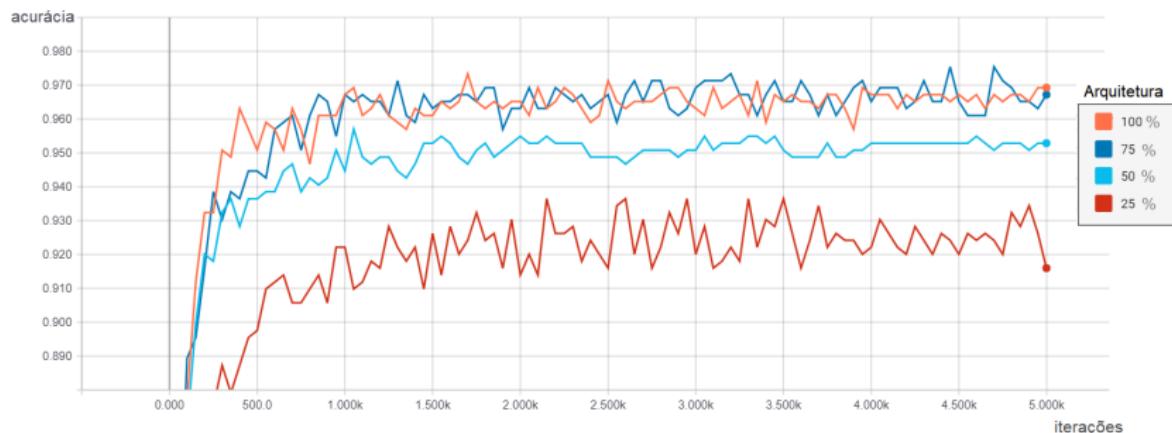


Figura: Acurácia das redes do grupo 2, no conjunto de validação, ao longo do treinamento.

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: Valores de acurácia, precisão, recall, F-score e tamanho das arquiteturas escolhidas.

Desenvolvimento

Resultados e Discussão

Aplicativo

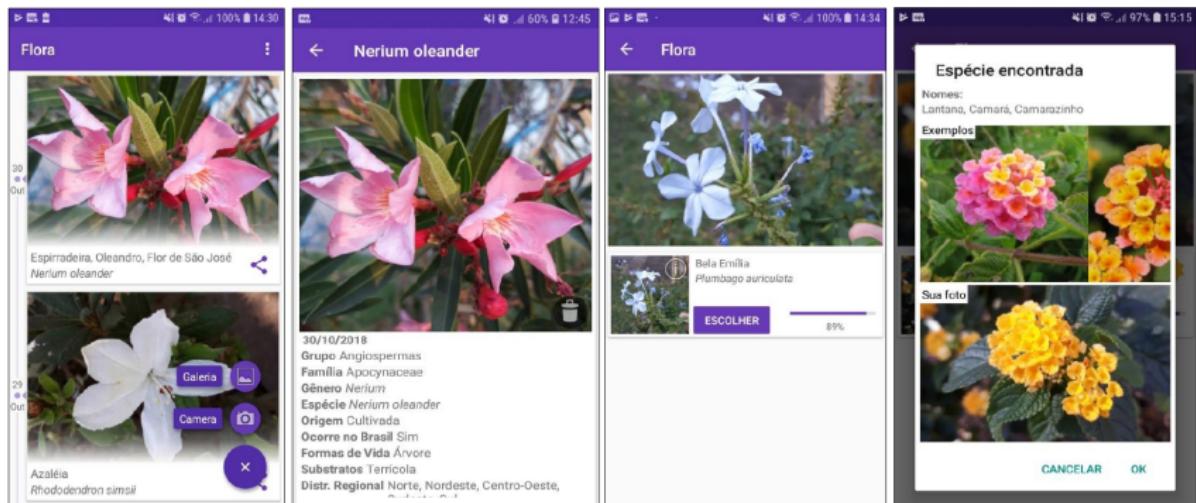


Figura: Capturas de tela do aplicativo. Fonte: elaborada pelo autor.

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

- [1] J. DENG, W. DONG, R. SOCHER, L.-J. LI, K. LI, AND L. FEI-FEI, *ImageNet: A Large-Scale Hierarchical Image Database*, in CVPR09, 2009.
- [2] A. G. HOWARD AND M. ZHU, *MobileNets: Open-Source Models for Efficient On-Device Vision*, 2017.
Disponível em: <https://ai.googleblog.com/2017/06/mobilenets-open-source-models-for.html>. Acessado em 3 de setembro de 2018.
- [3] A. G. HOWARD, M. ZHU, B. CHEN, D. KALENICHENKO, W. WANG, T. WEYAND, M. ANDREETTO, AND H. ADAM, *Mobilenets: Efficient convolutional neural networks for mobile vision applications*, CoRR, abs/1704.04861 (2017).

Referência Bibliográfica II

- [4] M.-E. NILSBACK AND A. ZISSERMAN, *102 Category Flower Dataset*, 2008.

Disponível em:

<http://www.robots.ox.ac.uk/~vgg/data/flowers/102/>.

Acessado em 10 de agosto de 2018.

- [5] M. SOKOLOVA AND G. LAPALME, *A systematic analysis of performance measures for classification tasks*, Information Processing & Management, 45 (2009), pp. 427 – 437.

Obrigado!

Dúvidas?

Aplicação de deep learning em dispositivos Android

Tiago de Miranda Leite

NUSP: 7595289

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