Food Recognition System for Nutrition Monitoring

Charles N. C. Freitass[1], Filipe R. Cordeiro[1], Adenilton J. da Silva[1]

[1]Departamento de Computação (DC)
Universidade Federal Rural de Pernambuco (UFRPE)

{charles.freitas,filipe.cordeiro, adenilton.silva}@ufrpe.br

Abstract. *This research consists of the analysis of the methods of image recognition, focusing on the problem of food classification, aiming to use the methods in a mobile application for the assistance in food monitoring and control. Thus, the development of the work contemplates the use of the deep learning method, focused on the recognition of food in images, with the use of neural convolution networks (CNN). For this purpose, a data set consisting of more than 1000 images and 5 food classes was constructed in order to simulate the SimpleNet, MiniVGGNet and Small Xception models, and thus define a learning model for food classification.*

Resumo. *Esta pesquisa consiste na análise dos métodos de reconhecimento de imagem, com foco no problema de classificação de alimentos, com objetivo de utilizar os métodos em uma aplicação móvel para o auxilio no monitoramento e controle alimentar. Assim, o desenvolvimento do trabalho contempla o uso do método de aprendizado profundo, voltados para o reconhecimento de alimentos em imagens, com a utilização das redes neurais convolutivas (CNN). Para esse fim, foi constituído um conjunto de dados com mais de 1000 imagens e 5 classes de alimentos, a fim de simular os modelos SimpleNet, MiniVGGNet e Small Xception, e assim definir um modelo de aprendizado para classificação de alimentos.*

1. Introdução

De acordo com o Instituto Brasileiro de Geografia e Estatística [IBGE 2010], a dieta brasileira não está totalmente balanceada para atingir parâmetros saudáveis. Pesquisa divulgada pelo Ministério da Saúde [SAÚDE 2016] revela que o índice de brasileiros acima do peso segue em crescimento no país. Isto é, mais da metade da população está nesta categoria (53,8%), e destes, 18,9% são obesos, porcentagem esta que cresceu cerca de 26,3% nos últimos dez anos. Este cenário, associado ao maior consumo de alimentos industrializados e à baixa prática de atividades físicas tem impacto direto na saúde dos brasileiros, contribuindo para uma alta incidência das doenças crônicas não transmissíveis, como diabetes e doenças cardiovasculares [SOUZA 2010].

A ausência de um monitoramento alimentar tem contribuído significativamente para o aumento do peso da população, e por conta disso acabam decidindo fazer uma dieta. Contudo, a falta de tempo e dinamicidade dos dias atuais não permite fazer o controle e o registro daquilo que se consome por meios de recursos tradicionais, como por exemplo, anotações, sistemas de pontos [1] e aplicativos de registro manual das refeições.

[1]Técnica usada para controlar a quantidade de alimentos ingeridos e realizar a substituição de forma inteligente. Com base em informações fornecidas pelo usuário, o sistema define a quantidade de pontos que se precisa para emagrecer sem prejudicar a saúde.

Assim, manter um padrão alimentar saudável torna-se um grande desafio para muitas pessoas. Contudo, a utilização cada vez mais recorrente de sistemas inteligentes tem auxiliado numa melhor adequação nutricional. Nos últimos anos, muitos trabalhos e pesquisas demonstraram que as técnicas de aprendizagem de máquina e visão computacional tem um alto potencial na construção de sistemas de reconhecimento automático de alimentos para estimar os valores nutricionais presentes em cada refeição. Contudo, a orientação de um médico ou nutricionista continua essencial para personalizar as recomendações de dietas, conforme alertam os especialistas [Gomes 2013].

Tomando por base os avanços tecnológicos e a crescente preocupação com a saúde e bem estar por parte da sociedade [Rodrigues 2013], tem-se como contribuição deste trabalho o desenvolvimento de um sistema de reconhecimento de alimentos para auxilio no monitoramento e controle alimentar. O desenvolvimento de um sistema automatizado e inteligente visa facilitar o registro dos alimentos e informar os usuários sobre o consumo dos nutrientes de cada refeição, atuando em contradição aos procedimentos tradicionais existentes, que em muitos casos são tarefas tediosas e demoradas [Kagaya et al. 2014].

O estudo de técnicas de deep learning para a classificação de alimentos ainda é pouco vista na literatura, o que torna esse trabalho significante para construção novas abordagem deste problema. Com isso a metotodologia adotada no projeto parte da análise de três arquiteturas, para definir um modelo preciso de reconhecimento de alimento, e assim aplicar ao sistema definido.

2. Trabalhos Relacionados

A aprendizagem profunda está fazendo grandes avanços na solução de problemas que resistiram às melhores tentativas da comunidade de inteligência artificial por muitos anos [Y. LeCun and Hinton 2015]. Um tipo particular de rede profunda, as redes neurais convolutivas (CNNs), tem ganhado destaque na comunidade científica por sua capacidade de generalização e treinamento mais facilitada. Trazendo grandes avanços na detecção, segmentação e reconhecimento de objetos e regiões em imagens [Y. LeCun and Hinton 2015].

No trabalho de Kagaya [Kagaya et al. 2014] *Food Detection and Recognition Using Convolutional Neural Network*, foi aplicado uma CNN para o reconhecimento de imagens alimentares. Esta abordagem, segundo o autor, mostra-se eficiente em relação aos métodos tradicionais de reconhecimento de imagem, por trabalhar com algoritmos de aprendizado de máquinas com menor ou nenhuma engenharia de características. Para realização dos experimentos foram adquiridas imagens de domínio alimentar, obtidas de um sistema de registro de alimentos, após foi aplicada a uma CNN para o reconhecimento de 10 classes de alimentos.

O reconhecimento de imagem de alimentos mostra-se uma área promissora no reconhecimento de objetos, uma vez que pretende-se estimar os valores nutricionais e a analisar os hábitos alimentares das pessoas para os cuidados com a saúde. Foi apresentado em Myers [Myers et al. 2016] um sistema que reconhece o conteúdo da refeição a partir de uma única imagem e, em seguida, prevê seus conteúdos nutricionais, como calorias. A versão mais simples pressupõe que o usuário está comendo em um restaurante pelo qual o cardápio é conhecido pelo sistema. Neste caso, pode-se coletar imagens offline, e treiná-las a partir de um classificador multi-rótulo. Para tal, os autores apresentaram uma

abordagem baseada em redes neurais convolutivas sobre esses problemas, com resultados preliminares promissores, que demonstraram uma precisão de classificação no conjunto de teste Food101 [Bossard et al. 2014] de 79%.

A estratégia desta pesquisa visa a utilização dos métodos de aprendizado profundo para geração de um modelo de aprendizado preciso no reconhecimento de imagens de alimentos. Tendo como objetivo principal a utilização do modelo gerado em aplicação móvel, possibilitando o registro dos alimentos consumidos durante o dia. Assim foram utilizados três arquiteturas do estado da arte relacionados a redes neurais convolutivas, para analisar qual melhor se adequá ao problema de pesquisa deste projeto. As arquiteturas utilizadas foram: *SimpleNet* (2016) [Chollet 2016], *MiniVGGNet* (2015) [Simonyan and Zisserman 2014] e *Small Xception* (2017) [Chollet 2017].

3. Aprendizagem Profunda

O aprendizado profundo ou *deep learning* refere-se a uma classe de redes neurais artificiais (RNAs) composta por muitas camadas de processamento [Moujahid 2016]. Uma rede neural simples possui apenas uma camada oculta (camadas intermediárias pelo qual é realizado o aprendizado do modelo), enquanto que uma rede profunda pode apresentar mais de uma camada oculta. Assim várias camadas ocultas permitem que as redes neurais profundas aprendam características dos dados por meio de uma hierarquia de conceitos [Goodfellow et al. 2016], permitindo que o computador aprenda conceitos complexos a partir da combinação de conceitos mais simples, como cantos e contornos, que são por sua vez definidos em termos de bordas [Goodfellow et al. 2016].

3.1. Redes Neurais Convolutivas

As Redes Neurais Convolutivas (ConvNets ou CNNs) são um tipo especial de redes *feed-forward*, que se mostraram muito eficazes em áreas como reconhecimento e classificação de imagens [Moujahid 2016]. A CNN é uma rede que contém vários tipos de camadas: convolutivas, pelo qual são realizados os primeiros filtros de processamento, camadas de agrupamento (*pool*) que permitem que a rede codifique certas propriedades de imagens, e por fim uma ou mais camadas totalmente conectadas com uma função de perda (por exemplo, SVM ou *Softmax*) para análise da rede. A saída no final pode ser uma única classe ou uma probabilidade de classes que melhor descrevem a imagem [Deshpande 2016].

As redes convolutivas possuem uma estrutura de filtros que mapeiam diferentes sinais em uma única imagem [Team 2017]. Elas utilizam esses filtros para mapear as características presentes em uma imagem. Ou seja, elas criam um mapa de cada lugar que essa característica ocorre. Ao aprender diferentes áreas de um espaço de característica, as redes convolutivas permitem criar uma mapa de características escalável e robusto.

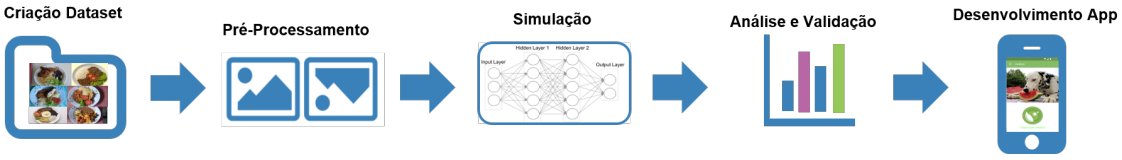
A arquitetura de uma CNN foi projetada para aproveitar a estrutura 2D de uma imagem de entrada (ou outra entrada 2D, como um sinal de fala) [Ng et al. 2013]. A entrada para uma camada convolucional é uma imagem ($m \times n \times r$) onde *m* é a altura e *n* largura da imagem, e *r* é o número de canais, uma imagem RGB possui *r* = 3, por exemplo. A camada convolucional terá *k* filtros (ou *kernels*) de tamanho $n \times n \times q$ onde *n* é a menor dimensão da imagem e *q* pode ser igual ao número de canais *r* ou menor, que pode variar para cada *kernel*. Assim, uma rede convolutiva recebe uma imagem colorida como matriz, cuja largura e altura são medidas pelo número de *pixels* ao longo dessas

dimensões e cuja profundidade é de três camadas profundas, uma para cada canal do RGB [Team 2017].

4. Sistema de classificação de alimentos

A estrutura do sistema foi definida em 5 etapas, que são definidas como: coleta e criação da base de dados; pré-processamento; implementação e simulação dos modelos; análise de resultados e desenvolvimento do app. Na Figura 1 é demonstrado o *pipeline* do ciclo de implementação do sistema de classificação de imagens de alimentos desenvolvido neste trabalho.

Figura 1. *Pipeline* do sistema de classificação de imagens de alimentos.



O foco principal está na implementação e simulação dos modelos escolhidos. E por fim a utilização do modelo que obteve melhores taxas de aprendizado em uma aplicação móvel.

4.1. Base de Imagens

O aprendizado profundo possui a capacidade de aprender características automaticamente a partir dos dados, o que geralmente só é possível quando muitos dados de treinamento estão disponíveis [Chollet 2016]. Para desenvolver um algoritmo que classifica imagens em categorias distintas, é necessário fornecer ao computador exemplos (imagens) de cada classe, para em seguida, processar esses exemplos e gerar uma aprendizagem sobre as características visuais de cada classe.

Esta abordagem é referida como abordagem baseada em dados, pelo qual consiste em um conjunto de dados de imagens rotuladas, que serão processadas e aprendidas através de um processo de treinamento. Uma vez que o modelo é treinado, ele pode ser testado usando um conjunto de dados independente para determinar quão bem ele pode generalizar para dados não vistos [Solomatine et al. 2008].

Para a constituição da base de imagens de alimentos utilizados no projeto, foi obtido a captura das imagens de três fontes distintas: Google Images com 275 imagens e Flickr com 1000 imagens. Resultando em um total de 1275 imagens, distribuídas em 5 (cinco) classes: Arroz (268), Feijão (252), Macarrão (282), Salada (291) e Carne Assada (182). Na Figura 2 é demonstrado um subconjunto de cada classe.

4.2. Pré-processamento e aumento de dados

Para aumentar os exemplos de treinamento, foi utilizado o modelo tradicional de transformações de imagens, que constitui, por exemplo, na combinação de operações de rotação, zoom e cortes nas imagens de entrada [Wang and Perez 2017]. Foram aplicadas em média 8 transformações para cada imagem de entrada, assim para um conjunto de dados N, geramos um conjunto de dados 8N. Dessa forma o conjunto de treinamento criado resultou em 2.000 instâncias de cada classe, totalizando 10.000 imagens. Um exemplo do resultado obtido a parti das transformações é demonstrado na Figura 3.

Figura 2. Imagens de alimentos da base utilizada.

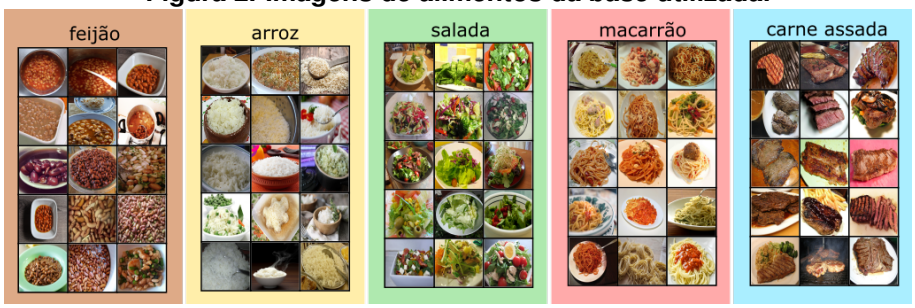


Figura 3. Transformação nas imagens para aumento de dados

4.3. Modelos Implementados

4.3.1. *SimpleNet*

SimpleNet é uma arquitetura constituída com poucas camadas de convolução e filtros. Foi proposta por Chollet [Chollet 2016] em seu artigo *Building powerful image classification models using very little data*, com a finalidade de construir um classificador de imagem simples e eficaz utilizando poucas amostras de treinamento. Por ser uma arquitetura simples, o artigo não denomina um nome para a arquitetura, assim por convenção foi adotado para este artigo, a denominação de *SimpleNet*.

A arquitetura do *SimpleNet* é composta por 3 camadas de convolução, todas com função de ativação *ReLU*, seguidas por camadas de agrupamento (*max-pooling*). Por fim, é utilizada uma camada de *flattening* (vetor de características usado pela camada densa para a classificação da rede) fora da rede, para em seguida uma camada totalmente conectada fazer a classificação. Para auxiliar na redução do *overfitting*, foi adicionada uma camada de *dropout*, impedindo que uma camada veja o mesmo padrão duas vezes, agindo de forma redundante aos dados, tornando o modelo independente de algumas entradas.

4.3.2. *MiniVGGNet*

A arquitetura da rede *VGG* foi introduzida por Simonyan e Zisserman em seu artigo de 2014, *"Very Deep Convolutional Networks for Large Scale Image Recognition"* [Simonyan and Zisserman 2014], sendo caracterizada pela sua simplicidade. Utiliza filtros de 3×3 nas camadas convolutivas empilhadas uma sobre a outra em profundidade crescente, reduzindo o tamanho do volume que é gerenciado pelo *max-pooling*. Duas camadas totalmente conectadas, cada uma com 4096 nós são seguidas por um classificador *softmax*. Para facilitar o treinamento os autores adotaram a estratégia de treinar versões menores do *VGG*, com menos camadas de peso. Com isso, as redes menores convergiam

e eram usadas como inicializações para redes maiores e mais profundas - esse processo é chamado de pré-treinamento.

Neste trabalho foi utilizada uma versão simplificada do modelo "*VGGNet*", chamada *MiniVGGNet*. O *MiniVGGNet* consiste em dois conjuntos de camadas (CONV $=>$ RELU $=>$ CONV $=>$ RELU $=>$ POOL), seguido de um conjunto de camada (FC $=>$ RELU $=>$ FC $=>$ SOFTMAX). Camadas de abandono (*dropout*) foram adicionadas entre cada conjunto de camadas POOL e FC, a utilização do *dropout* é necessária para reduzir as chances de *overfitting* e aumentar a precisão do modelo.

4.3.3. *Small Xception*

O *Xception* foi proposto por François Chollet [Chollet 2017], criador e chefe da biblioteca Keras. O *Xception* é uma extensão da arquitetura *Inception* que substituí os módulos padrões de inicialização por contornos separáveis. A proposta da arquitetura *Xception*, segundo o autor, é constituir uma rede neural convolutiva baseada inteiramente em camadas de convolução separáveis em profundidade. A arquitetura possui 36 camadas convolutivas, formando a base de extração de recursos da rede, que são estruturadas em 14 módulos, todos com conexões residuais lineares em torno delas, com exceção do primeiro e último módulo [Chollet 2017], tendo assim, uma utilização mais eficiente dos seus parâmetros, influenciando em bons ganhos de desempenho.

A arquitetura *Xception* é uma pilha linear de camadas de convolução separáveis em profundidade com conexões residuais [Chollet 2017]. Isso torna a arquitetura muito fácil de definir e modificar. Para este trabalho, devido ao esforço computacional e tempo de simulação para o mesmo, foi utilizado uma versão simplificada do do modelo "*Extreme Inception*" ou *Xception* original, denominada *Small Xception*, pelo qual não são utilizadas conexões residuais.

Em todos os modelos criados na última camada (camada de saída), foram utilizados a função de ativação *Softmax*, que é a função responsável por definir a saída de cada neurônio, de acordo com a classe correspondente, assumindo um padrão de saída codificado em uma distribuição de probabilidades sobre as 5 classes de saída definidas nos modelos implementados.

4.4. Avaliação dos Modelos

Para este projeto foi adotado a utilização de 5 *folds* da base de imagens utilizada, distribuídos da seguinte forma: 60% das instâncias para treino, 20% para validação e 20% para teste. A utilização de 5 *folds* ocorre pelo fato do custo de tempo de execução e avaliação de cada modelo de aprendizagem. Foram utilizadas 50 épocas -o número de épocas foi reduzido devido ao custo computacional- e tamanho de batch de 64.

Utilizamos a função de perda logarítmica, para o problema de classificação multiclasse. Para a otimização foi utilizado o algoritmo de gradiente descendente estocástico, ou sgd, que requer a ajuste na taxa de aprendizado e momento, que neste caso foram definidos com valores de 0.01 e 0.9 respectivamente. E por fim, a especificação da métrica de avaliação do modelo criado, pelo qual foi utilizado a acurácia e função de perda do modelo de cada classificação.

A precisão foi calculada em todos os rótulos, medindo o número de vezes que uma determinada classe foi predita corretamente (verdadeiros positivos). A precisão determinada pelo rótulo considera apenas uma classe e mede o número de vezes que um rótulo específico foi predito corretamente, normalizado pelo número de vezes que o rótulo aparece na saída. Na Equação 2 é demonstrado o cálculo da acurácia, onde vp são os rótulos verdadeiros positivos e fp os rótulos falsos positivos.

$$acc = \frac{vp}{vp + fp} = \frac{1}{N} \sum_{i=0}^{N-1} \widehat{\delta}(\widehat{y_i} - y_i) \tag{1}$$

Definimos a classe como $L = l_0, l_1, ..., l_{m-1}$, assim o algoritmo de predição multi classe gera um vetor de saídas verdadeiras y, que consiste de N elementos $(y_0, y_1, ..., y_{n-1})\epsilon L$, e um vetor de predição $\widehat{y}$ de N elementos $(\widehat{y}_0, \widehat{y}_1, ..., \widehat{y}_{N-1})\epsilon L$, pelo qual é determinado os rótulos que foram classificados corretamente, que são definidos pela função delta $\widehat{\delta}(x)$, sendo calculado da seguinte forma $\widehat{\delta}(x) = \begin{cases} 1 \text{ se } x = 0, \\ 0 \text{ otherwise.} \end{cases}$

Na função de perda é expressa a diferença entre as previsões do modelos treinados, e as verdadeiras instâncias do problema. Seu objetivo é designar a imprecisão das previsões realizadas em nossos modelos treinados. A função de perda utilizada foi a *categorical_crossentropy*, que retorna a entropia cruzada entre duas distribuições de probabilidade, que mede o número médio de bits necessário para identificar um evento a partir de um conjunto de possibilidades. Matematicamente, esta função é calculada conforme a Equação 2, onde p é conjunto de rótulos verdadeiros, q é o conjunto de previsões, y o rótulo verdadeiro e $\widehat{y}$ o rótulo previsto.

$$H(p, q) = -\sum p_i \log q_i = -y \log \widehat{y} - (1 - y) \log(1 - \widehat{y}) \tag{2}$$

5. Resultados e Discussões

Para os três modelos simulados no projeto: *SimpleNet*, *Small Xception* e *MiniVGGNet*, foram avaliados o desempenho (acurácia e taxa de perda) de cada época dos subconjuntos (*folds*) constituídos. Calculamos a média e desvio padrão das melhores épocas de cada *fold*, para ter uma estimativa da melhor combinação realizada no processo de validação cruzada para o modelo implementado.

Na figura 4 é apresentado todas as média calculadas com base na validação cruzada sobre a taxa de acurácia do subconjunto de validação. O modelo de aprendizado *SimpleNet* obteve média geral de 74,46% e desvio padrão de 0,01%, o modelo *Small Xception* obteve média geral de 86,94% e desvio padrão de 0,01%, o modelo *MiniVGGNet* obteve média geral de 72,09% e desvio padrão de 0,01%.

A Tabela 1 apresenta as estatística em relação as métricas de acurácia e perda do conjunto de treinamento e teste. É demonstrando também as médias e desvio padrão, correspondentes a cada modelo.

Figura 4. Média geral acurácia subconjunto de validação

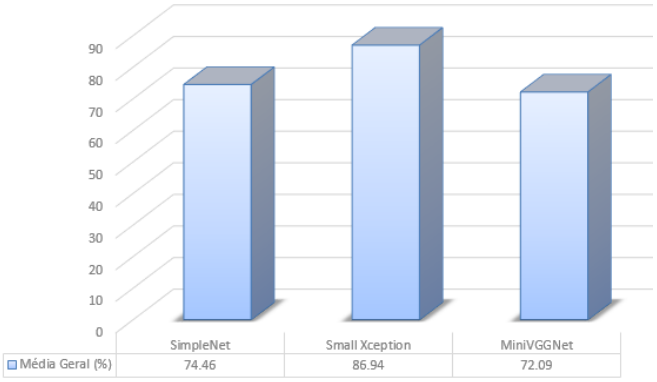


Tabela 1. Resumo da avaliação dos melhores modelos

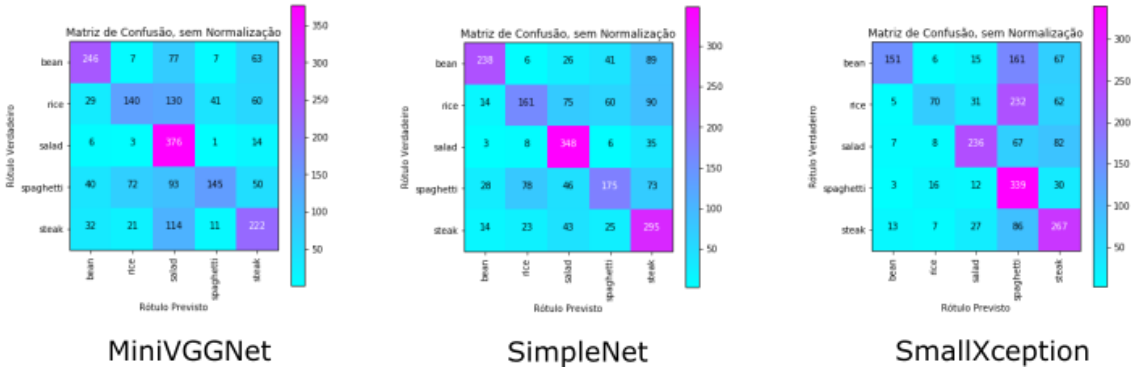
			Treino			Validação		
Modelo	**Fold**	**Época**	**Acurácia**	**Perda**	**Média**	**Acurácia**	**Perda**	**Média**
SimpleNet	4	31	83,04	44,25	74,75 (0,12)	76,10	67,08	71,59 (0,07)
SmallXception	4	38	98,15	06,10	90,67 (1,13)	88,60	37,31	79,62 (1,37)
MiniVGGNet	2	15	79,91	52,14	82,14 (0,14)	73,68	74,98	70,84 (0,06)

5.1. Análises das Predições

Com base nos modelos apresentados em cada arquitetura simulada, foram realizadas algumas análises relacionadas ao conjunto de teste, para saber qual melhor modelo corresponde ao problema de classificação de alimento, proposto neste protejo. Para tal, foi gerado a matriz de confusão e uma análise preditiva sobre novos dados do conjunto de teste para cada modelo, correspondendo a 400 imagens de teste, para cada classe: Feijão, Arroz, Salada, Macarrão e Carne Assada, totalizando 2000 (duas mil) imagens de teste.

Em todas as arquiteturas foram utilizados os melhores modelos analisados nas simulações para a realização dos testes de predição. Na Figura 5 é demonstrando as matrizes de confusão sobre a quantidade de acertos e erros para cada classe.

Figura 5. Matrizes de confusão dos modelos implementados

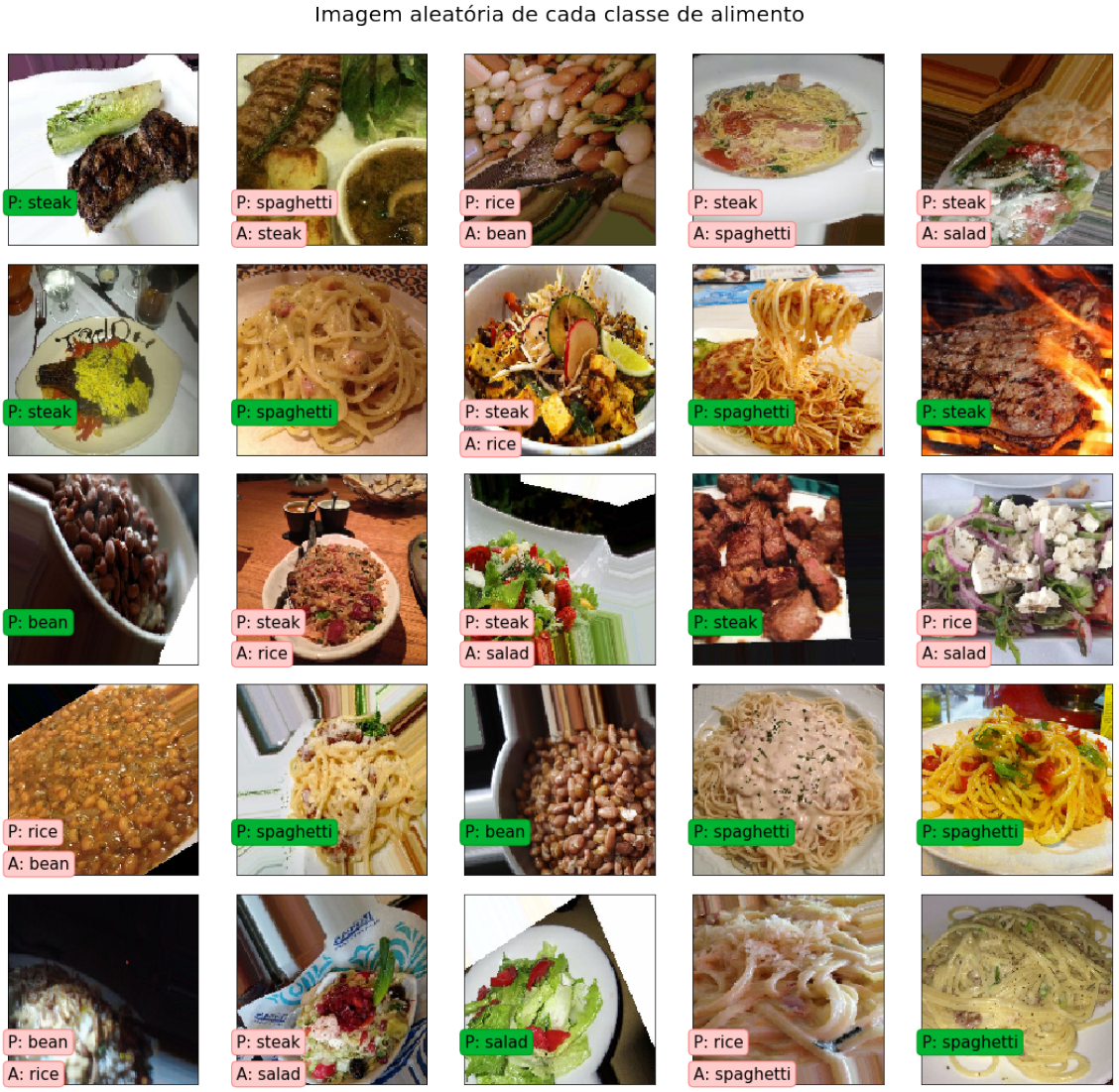


Na arquitetura *SimpleNet* as classes que obtiveram maior precisão na classificação de novos dados foram: salada (348), carne (295) e feijão (238). Na arquitetura *Small Xception* as classes que obtiveram maior precisão na classificação de novos dados foram:

macarrão (339), carne (267) e salada (236). E na arquitetura *MiniVGGNet* as classes que obtiveram maior precisão na classificação de novos dados foram: salada (376), feijão (246) e carne (222).

Na Figura 6 é apresentado um exemplo de uma matriz de predição de imagens do modelo *SimpleNet*, pelo qual foram escolhidas aleatoriamente 25 imagens da base de teste, demonstrando a classificação gerada pelo modelo. A letra P apresentada na imagem representa o rótulo da imagem prevista pelo modelo, a letra A representa o rótulo verdadeiro correspondente a imagem. É possível observar que o modelo classificou corretamente $13/25$ das imagens selecionadas, tendo para essa amostra um total de 52% de acerto na previsão. A classe feijão teve maior quantidade de acerto $6/13$, representando, aproximadamente 46% das imagens classificadas corretamente.

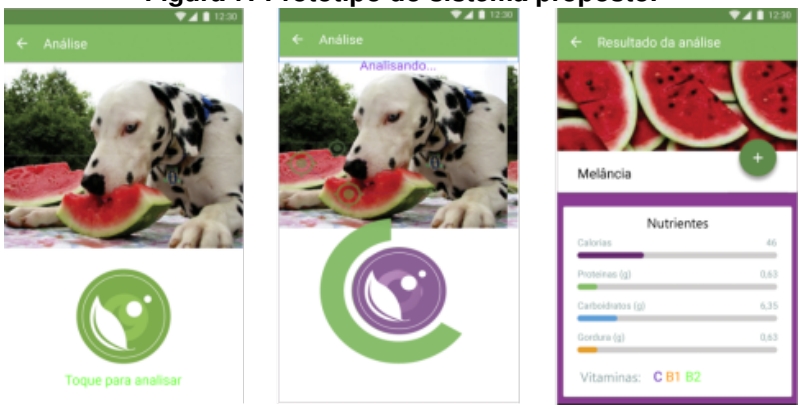
Figura 6. Predição de 25 imagens aleatórias de cada classe de alimento do modelo *SimpleNet*.



5.2. Aplicação do modelo

Após a definição e análise dos modelos a última etapa é o desenvolvimento da aplicação mobile e utilização do modelo que obteve melhores taxas de precisão no reconhecimento de novas instâncias. A Figura 7 é demonstrado o protótipo desenvolvido com o intuito de concretizar a ideia geral do sistema proposto, focando-se na boa usabilidade, visto que é uma medida de qualidade que avalia o quão fácil é utilizar uma interface na visão do usuário. Dessa forma, quanto melhor a usabilidade de uma interface, mais fácil será de utilizá-la, por isso são essenciais no desenvolvimento de software [Nielsen 2012] e [Ferreira 2014].

Figura 7. Protótipo do sistema proposto.



A aplicação desenvolvida visa além do registro automático de imagens de alimentos e fornecimento das estimativas nutricionais, a interação dos usuários com os profissionais de nutrição para melhor adequação das dietas e obtenção de informações nutricionais que podem contribuir para reeducação alimentar.

6. Conclusão e Trabalhos Futuros

Este trabalho teve como foco o desenvolvimento de um sistema computacional de classificação de alimentos, através do reconhecimento de imagem. O sistema proposto é capaz de reconhecer cinco classes de alimentos, com a utilização das técnicas e métodos de aprendizado profundo. Foram realizados testes e validações para três modelos de aprendizado, com taxa de precisão na média geral, do estado da arte, para este problema de classificação de imagem.

Os modelos implementados obtiveram boas taxas de aprendizagem em relação ao tamanho do conjunto de dados e as arquiteturas implementadas. Assim podemos concluir que a metodologia implementada nestes modelos, para classificação de imagens de alimentos, mostra-se uma abordagem significativa para o reconhecimento de padrões em imagens.

Como proposta de melhorias e trabalhos futuros será realizado o processo de segmentação das imagens, para ter um reconhecimento mais especializado, focado no objeto principal da imagem, tendo assim respostas mais precisa na detecção de alimentos. Além disso, será realizado a continuação do desenvolvimento da aplicação móvel visando melhorar suas funcionalidades em termos de usabilidade e reconhecimento de novas classes de alimentos.

7. Agradecimentos

Este trabalho recebeu apoio do Instituto Serrapilheira (número do processo Serra-1709-22626)

Referências

[Bossard et al. 2014] Bossard, L., Guillaumin, M., and Van Gool, L. (2014). Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*.

[Chollet 2016] Chollet, F. (2016). Building powerful image classification models using very little data. Available at https://blog.keras.io/building-powerful-image-classification-models-using- very-little-data.html.

[Chollet 2017] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions.

[Deshpande 2016] Deshpande, A. (2016). A beginner's guide to understanding convolutional neural networks. Available at https://adeshpande3.github.io/adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/.

[Ferreira 2014] Ferreira, B. M. (2014). Usabilicity: um jogo de apoio ao ensino de propriedades de usabilidade de software através de analogias. Technical report, XXV Simpósio Brasileiro de Informática na Educação – SBIE.

[Gomes 2013] Gomes, M. (2013). De dieta em dieta: o que a ciência diz sobre as soluções milagrosas. ComCiência, Campinas. Available at http://comciencia.scielo.br.

[Goodfellow et al. 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. `http://www.deeplearningbook.org`.

[IBGE 2010] IBGE (2010). Censo do instituto brasileiro de geografia e estatística. Available at http://censo2010.ibge.gov.br.

[Kagaya et al. 2014] Kagaya, H., Aizawa, K., and Ogawa, M. (2014). Food detection and recognition using convolutional neural network.

[Moujahid 2016] Moujahid, A. (2016). A practical introduction to deep learning with caffe and python. Available at http://adilmoujahid.com/posts/2016/06/introduction-deep-learning-python-caffe/.

[Myers et al. 2016] Myers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., Guadarrama, S., Papandreou, G., Huang, J., and Murphy, K. (2016). Im2calories: towards an automated mobile vision food diary.

[Ng et al. 2013] Ng, A., Ngiam, J., Foo, C. Y., Mai, Y., Suen, C., Coates, A., Maas, A., Hannun, A., Huval, B., Wang, T., and Tandon, S. (2013). Deep learning tutorial. Technical report, Stanford.

[Nielsen 2012] Nielsen, J. (2012). Usability 101: Introduction to usability. Available at http://www.nngroup.com/articles/usability-101-introduction-to-usability.

[Rodrigues 2013] Rodrigues, P. M. P. (2013). Reconhecimento automático de calorias numa refeição. Master's thesis, Faculdade de Ciências da Universidade do Porto.

[SAÚDE 2016] SAÚDE, M. D. (2016). Hábitos dos brasileiros impactam no crescimento da obesidade e aumenta prevalência de diabetes e hipertensão. Technical report, VIGITEL BRASIL.

[Simonyan and Zisserman 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *Visual Geometry Group, Department of Engineering Science, University of Oxford.*

[Solomatine et al. 2008] Solomatine, D., See, L. M., and Abrahart, R. J. (2008). Data-driven modelling: Concepts, approaches and experiences. *Springer-Verlag Berlin Heidelberg*.

[SOUZA 2010] SOUZA, E. B. (2010). Transição nutricional no brasil: análise dos principais fatores. *Cadernos UniFOA. Volta Redonda, Ano V, n. 13*.

[Team 2017] Team, D. D. (2017). Deeplearning4j: Open-source distributed deep learning for the jvm, apache software foundation license 2.0. Available at http://deeplearning4j.org.

[Wang and Perez 2017] Wang, J. and Perez, L. (2017). The effectiveness of data augmentation in image classification using deep learning. *CORNELL UNIVERSITY LIBRARY*.

[Y. LeCun and Hinton 2015] Y. LeCun, Y. B. and Hinton, G. (2015). Deep learning. *NATURE — VOL 521*.