

Utilizando redes convolucionais para classificação de pratos vegetarianos

Leonardo Maldonado
172017
l172017@dac.unicamp.br

Gabriel Costa
172269
g172269@dac.unicamp.br

Giovanni Bertão
173325
g173325@dac.unicamp.br

Rodrigo Freitas
176848
r176848@dac.unicamp.br

Resumo—Há diversas restrições alimentares presentes na dieta de diversas pessoas. A adoção de uma ferramenta que possa classificar em qual grupo cada refeição se encaixa é interessante para a avaliação de contemplação de restrições. Este trabalho propõe e avalia uma forma automatizada de identificar um prato como vegetariano ovo-lacto, vegetariano estrito ou nenhum dos anteriores. Esse classificador utiliza redes neurais convolucionais e obteve acurácia de aproximadamente 50% para identificar pratos vegetarianos estritos, vegetariano ovo-lacto e pratos com carne.

I. INTRODUÇÃO

As restrições alimentares baseadas na origem dos ingredientes têm se tornado cada vez mais frequente por diversos motivos e valores [1]. Além desses aspectos, há a combinação de interesses individuais e sociais quanto à saúde relacionada à alimentação. Isso porque a qualidade das refeições está diretamente associada ao estado de bem-estar físico, mental e psicológico. Nesse contexto, sabe-se que o consumo elevado de carne está associado à doenças crônicas cardiovasculares e grandes impactos ambientais [2]. Dessa forma, as dietas que restringem o consumo de ingredientes de origem animal são cada vez mais praticadas [3] e estão associadas à melhora na saúde geral dos praticantes [4].

Dada essa transformação cultural e social no hábito alimentar [3], motivada por diversos fatores, surge a necessidade de se avaliar o pertencimento de um determinado prato nas restrições adotadas. Neste trabalho foram escolhidas as categorias **vegetariano estrito**, correspondente à refeições cujos ingredientes não contém nenhuma origem animal, **vegetariano ovo-lacto**, que restringe o consumo de origem animal a leite, ovos e derivados e a dieta **livre**, que inclui qualquer ingrediente que seja do interesse nutricional e palato-gustativo do indivíduo, isto é, incluindo o consumo de carne.

Como a adoção de restrições que cabem dentro dessas divisões é relativamente frequente, é de grande praticidade a construção de um classificador que possa associar imagens de comida à categoria correspondente. Desse modo, é possível, por exemplo, contar o número de pratos vegetarianos que um restaurante oferece de forma automática, avaliando as imagens apresentadas em seu cardápio. Além disso, esse tipo de *software* pode auxiliar no cumprimento da restrição, avaliando se um alimento contempla ou não uma determinada dieta.

Vale notar que, apesar dos avanços na classificação de imagens, a associação de alimentos à categorias ainda é um

processo razoavelmente mais difícil do que a classificação geral de imagens [5].

A Seção II apresenta trabalhos similares que utilizaram técnicas de aprendizagem de máquina para classificação de alimentos. A Seção III contém a descrição da solução proposta, reportando quais dados foram utilizados neste trabalho, como os mesmos foram obtidos e as técnicas aplicadas. Por fim, a Seção IV comenta sobre as conclusões e dificuldades encontradas ao se trabalhar com classificação de alimentos.

II. ATIVIDADES RELACIONADAS

Há diversos trabalhos que propuseram e investigaram o uso de *software* para avaliar e classificar imagens de pratos e refeições sob os mais diversos aspectos. Tiankawe et al. [6] construíram um aplicativo para *smartphones* que utiliza uma rede neural convolucional para estimar a quantidade de calorias presentes em 13 pratos tailandeses diferentes a partir de uma foto da refeição.

Singla et al. [7] comentam a importância do uso de classificadores de imagem para aumentar a qualidade do processo de avaliação dietética e propõem o uso de redes convolucionais para identificar alimentos presentes em fotos.

Zhu et al. [8] reportaram bons resultados classificando o tipo e identificando a posição de alimentos em fotos de dispositivos móveis. Para isso, foi utilizado máquina de vetores de suporte para classificação e corte normalizado para segmentação (*normalized cuts*).

Ragusa et al. [9] obtiveram boa acurácia na classificação binária de imagens nas categorias comida e não-comida. Entre outras aplicações, esse trabalho é importante, por exemplo, na construção de *datasets* de alimentos, permitindo a verificação de existência de imagens inválidas.

A maioria dos trabalhos supracitados comenta a dificuldade existente na classificação de alimentos, além de citar a importância da análise nutricional e composição de ingredientes para a saúde humana, ressaltando os pontos positivos de se realizar esse exame de forma automatizada [6]–[9]. Estes últimos utilizaram métodos supervisionados similares aos propostos neste trabalho, mas é importante notar que há outros procedimentos na literatura que obtiveram resultados satisfatórios para problemas análogos [10].

III. SOLUÇÃO PROPOSTA E EXPERIMENTOS

Dado o interesse em se construir um classificador que identifique, baseado em uma imagem, a qual grupo pertence

um dado alimento, expresso na Seção I, neste trabalho é proposta uma rede convolucional que utiliza a foto de um alimento para identificá-lo como vegetariano estrito, ovo-lacto-vegetariano ou carnívoro.

Todas as arquiteturas propostas e examinadas foram construídas utilizando a API do Keras [11], com Python 3.7 e TensorFlow [12]. Estas então foram executadas no *Google Colab* [13] com o uso de GPU para aceleração, dispositivo que apresenta bom desempenho para treino de redes convolucionais [14], [15].

A. Dataset

A qualidade e potencial de generalização de um classificador dependem fortemente dos dados utilizados em seu treino. Sendo assim, após os experimentos iniciais com o *dataset Food101*, foram avaliados alguns acréscimos de dados.

O *Food101*, disponível no *Kaggle*¹ contém imagens de pratos de dimensão 64x64 com três canais de cor. Nos exemplos presentes está associada uma anotação com o nome do prato. Sendo assim, os dados foram anotados novamente para contemplar as três classes esperadas nesse trabalho. O processo de re-anotação foi feito cruzando a receita com o nome do alimento e está ilustrado na Figura 1.

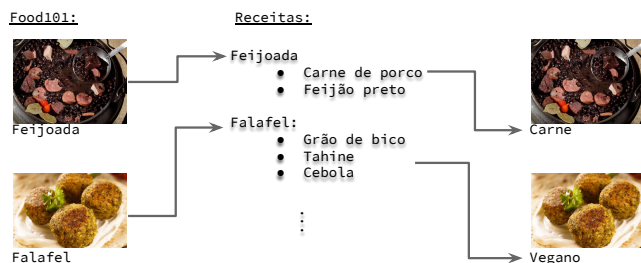


Figura 1. Ilustração do processo de re-anotação do *dataset*.

Além disso, para melhorar os resultados e tornar o modelo mais generalizável, foram adicionadas novas imagens, aumentando de cerca de 900 exemplos de pratos vegetarianos estritos para 2000. Isso foi feito utilizando *web crawlers* para acessar *sites* de receitas e realizar o *download* das imagens. Essa adição tem grande importância, pois, dada a variedade de análises que podem ser feitas sobre comida, a divisão por características específicas pode tornar o *dataset* muito desbalanceado, sendo muitas vezes necessário acrescentar dados ou construir o próprio conjunto nesse tipo de trabalho [5], [7].

B. Baseline: CNN para Food101 re-anotado

Para o primeiro experimento, foi utilizado o *dataset Food101* mapeado para as três classes propostas e alimentadas em uma CNN, feita de duas camadas de convoluções, uma camada de *max pooling*, uma camada *flatten* e duas camadas densas (128 e 3 nós, respectivamente) com *dropout* de 20% após cada camada, como ilustra a Figura 2. Essa configuração apresenta um total de 2,151,363 parâmetros.

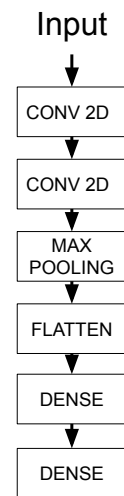


Figura 2. CNN utilizada para experimentos iniciais (*baseline*).

O resultado inicial, utilizando o *Food101* apenas modificando-se as *labels* para as três classes propostas, teve acurácia de aproximadamente 55% no conjunto de validação. Apesar de ser um número razoável, ele não representa um bom resultado, pois o *dataset* estava desbalanceado para a nova anotação, com a grande maioria dos pratos contendo carne. Nesse teste a rede aprendeu a classificar toda entrada como "contendo carne". Como esse tipo de dado era maioria, o resultado aparenta ser muito melhor do que de fato é.

C. Balanceando o Dataset

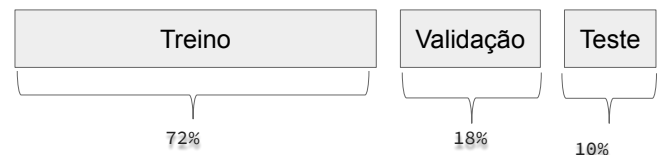


Figura 3. Divisão dos dados utilizada em todos os experimentos, exceto quando indicado o contrário. Porcentagens correspondem à proporção de exemplos em cada conjunto para o *dataset* balanceado com os dados adicionados com os meios descritos. O conjunto de treino é utilizado para calibrar os parâmetros e a validação para se verificar a acurácia e custo ao longo das épocas. O conjunto de teste foi utilizado apenas no final para se obter a matriz de confusão de cada modelo.

Para balancear o *dataset*, foi utilizado um site de agregador de receitas² como fonte de imagens de alimentos que seguem a dieta vegetariano estrito.

Este site junta receitas de diversos regimes, utilizando *links* de páginas de terceiros que escrevem a lista de ingredientes, o modo de preparo e enviam as fotos. A vantagem deste levantamento de dados deve-se ao fato de que a validação do rótulo dado para a imagem se encontra na receita que o autor coloca junto à figura correspondente. Dessa forma, bastou apenas a

¹<https://www.kaggle.com/kmader/food41>

²<https://www.yummly.com>

leitura dos ingredientes em busca de algum componente que quebrasse as regras para uma dieta vegetariana estrita, e, caso não ocorresse nenhuma quebra de regra, se adicionava ao grupo compatível do *dataset* no mesmo formato h5 proposto pelo *dataset Food101* redimensionadas ao formato 64x64 em RGB.

D. Utilizando outra CNN para o dataset balanceado para as 3 classes

Utilizando o *dataset* balanceado com 3891 instâncias de imagens, foram avaliadas diversas configurações diferentes para uma CNN. Primeiro, foi testado um modelo formado por três camadas convolucionais bidimensionais (tamanho de *kernel* 3, 20 e 3, respectivamente, e com função de ativação ReLU), uma camada *flatten*, duas camadas densas separadas com *dropout* de 20% e a camada de saída, também *fully connected* com três neurônios. Nesses experimentos, variou-se a função de ativação das camadas em busca de melhores resultados, com o uso de funções como *softmax* e *sigmoid*.

Para compilar o modelo, variou-se o otimizador utilizado (a saber, Adam, SGD e Adagrad), para verificar qual tinha melhor impacto na convergência. A função de custo utilizada foi a entropia cruzada categórica e o erro quadrático médio. As CNNs avaliadas foram comparadas por sua acurácia e função de custo ao longo das épocas.

Nesses experimentos, grande parte das redes apresentou baixa acurácia, nos quais se verificou que o modelo não convergia para a inferência esperada. Para tais ocorrências, a análise da matriz de confusão mostrou que a saída da rede era constante para todas os exemplos testados, como mostra a Figura 4. Como resultado, a acurácia foi de apenas 33%, aproximadamente. Esse comportamento indesejado pode ser observado nas Figuras 5 e 6, nas quais a variação do custo e da acurácia com a passagem épocas é indesejadamente errática, indicando não convergência da rede.

Vegano	334	0	0
Ovo-lacto	336	0	0
Carne	303	0	0
	Vegano	Ovo-lacto	Carne

Figura 4. Exemplo de matriz de confusão da CNN inicial que não convergiu.

Apesar disso, alguns testes apresentaram resultados satisfatórios: outra CNN com três camadas convolucionais, uma *flatten* e três camadas densas teve acurácia de aproximadamente 45%. Para essa rede, é possível notar na matriz de confusão (Figura 7) que as classificações incorretas estão

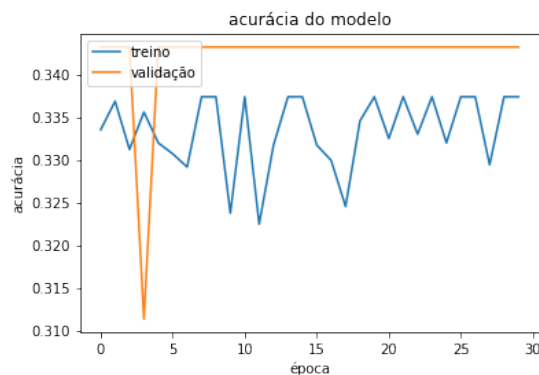


Figura 5. Acurácia por épocas para CNN inicial que não convergiu.

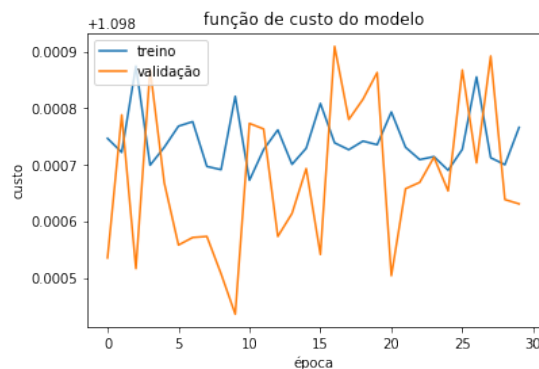


Figura 6. Custo por épocas para CNN inicial que não convergiu.

distribuídas de maneira mais coerente com os exemplos. Sendo assim, é possível observar a convergência do modelo com a passagem das épocas nas Figuras 8 e 9.

Vegano	150	88	83
Ovo-lacto	74	138	107
Carne	83	144	106
	Vegano	Ovo-lacto	Carne

Figura 7. Exemplo de matriz de confusão da CNN padrão com resultados interessantes.

E. Transfer learning com a VGG

Há diversos trabalhos que utilizaram *transfer learning* para obter melhora na acurácia de classificadores, inclusive para classificação de alimentos [7]. Esta técnica consiste em utilizar uma rede que já foi treinada para classificar um novo conjunto

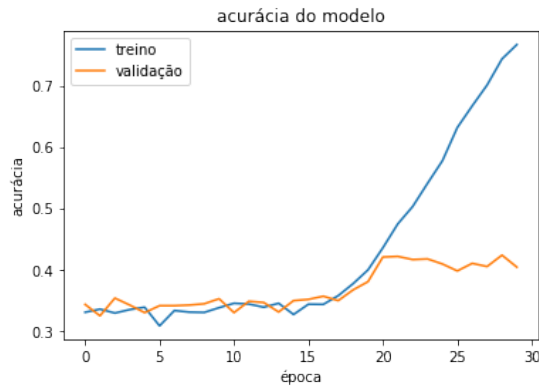


Figura 8. Exemplo de função de acurácia para as épocas da *CNN* padrão com resultados interessantes.

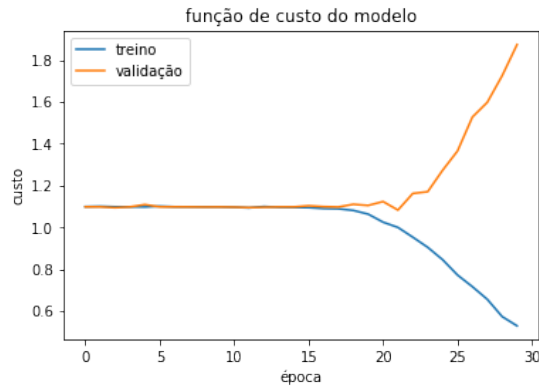


Figura 9. Exemplo de função de Custo para as épocas da *CNN* padrão com resultados interessantes.

de dados, neste caso, foi utilizada a VGG16, que foi treinada a partir do conjunto de dados da competição *ImageNet*.

Com isso, não é preciso treinar a rede novamente para o novo conjunto de dados. Isso porque os pesos das camadas escondidas são os obtidos com o treinamento da rede com *dataset* da competição, de forma que é calibrada apenas a última (ou últimas) camadas, que são substituídas por camadas que façam sentido para o novo conjunto de dados. Desse modo, mesmo que o conjunto de dados utilizado não seja tão grande, pode-se obter um resultado satisfatório, pois um conjunto suficientemente grande de imagens foi utilizado no treino.

Com o uso dessa técnica na VGG16 para o *dataset* proposto, o resultado foi consideravelmente superior ao dos experimentos anteriores, como ilustra a Figura 10, indicando que essa metodologia é promissora para o problema tratado.

F. Data Augmentation

Devido à quantidade limitada de dados para o experimento, foi utilizado *data augmentation*, que aplica alterações que não descaracterizam a imagem, mantendo-a consistente com sua anotação, mas gerando versões modificadas. Desse modo, a quantidade de exemplos disponíveis para o treinamento da rede é maior.

Vegano	188	76	57
Ovo-lacto	51	179	89
Carne	62	159	112
	Vegano	Ovo-lacto	Carne

Figura 10. Matriz de confusão para o uso de *transfer learning* com a VGG16.

Foram aplicadas as seguintes aumentações:

- Deslocamento horizontal dos pixels.
- Deslocamento vertical dos pixels
- Rotação de até 90 graus.
- Inversão horizontal da imagem.
- Inversão vertical da imagem.
- Modificação no brilho.
- Modificação no *zoom*.

Na Figura 11 é possível conferir diversas aumentações aplicadas a uma imagem do *Food101*.

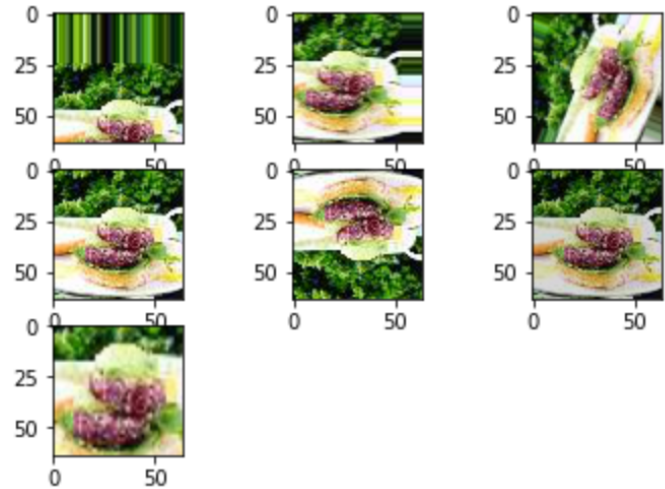


Figura 11. Exemplos de *data augmentation*.

G. CNN inicial com Data Augmentation

Para avaliação do impacto de cada técnica no caso estudado, estas foram combinadas com as arquiteturas de rede propostas. No primeiro teste, foi utilizado o *dataset* balanceado acrescido de novas imagens, com *data augmentation*, na CNN inicial. Além disso, foram aplicadas diferentes aumentações separadamente, mensurando-se seu impacto.

Nos experimentos feitos com a mesma arquitetura da CNN inicial e com o uso de *horizontal flip*, aumento na qual a imagem é invertida horizontalmente, observa-se na Figura 12

a evolução da acurácia ao longo das épocas no conjunto de treino e de validação. A Figura 13 mostra a relação da *loss* com o passar das épocas. Analisando essas figuras é possível notar a ocorrência de *overfitting* após a passagem de mais de cinco épocas.

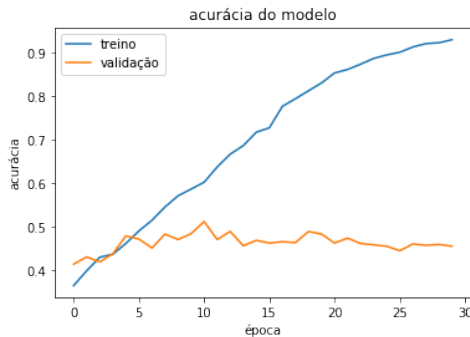


Figura 12. Relação da acurácia através das épocas na CNN inicial usando *horizontal flip*.

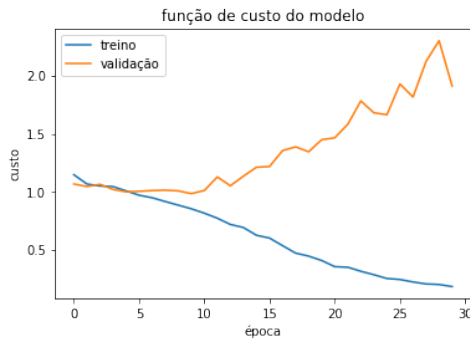


Figura 13. Relação da *loss* através das épocas na CNN inicial usando *horizontal flip*.

Analogamente, foi feito o mesmo teste substituindo o uso de *horizontal flip* por *vertical flip*. Nessa técnica, a imagem é invertida verticalmente. Como é uma aumentação similar, os resultados foram parecidos, conforme esperado. Diante disso, obteve-se os resultados expressos nas Figuras 14 e 15, que mostram novamente o *overfitting* após cerca de cinco épocas.

Outro teste desempenhado na CNN inicial foi a combinação das duas técnicas de aumentação de dados citadas. Os resultados podem ser observados na Figura 16 que mostra a evolução da acurácia ao longo das épocas do conjunto de treino e de validação. Adicionalmente, a Figura 17 mostra a relação da *loss* com o passar das épocas. Novamente, os resultados demonstram o *overfitting* do modelo. Isto é, mesmo a combinação das técnicas não foi muito expressiva para melhorar a generalização do modelo.

Outra forma de *Data Augmentation* experimentada foi a rotação das imagens em graus arbitrários entre 0 e 90. Com essa técnica, diferentemente do espelhamento, muitas variações podem ser geradas para cada exemplo. O uso dessa aumentação na CNN inicial pode ser visto no *plot* da Figura 18 que mostra a evolução da acurácia ao longo das épocas no

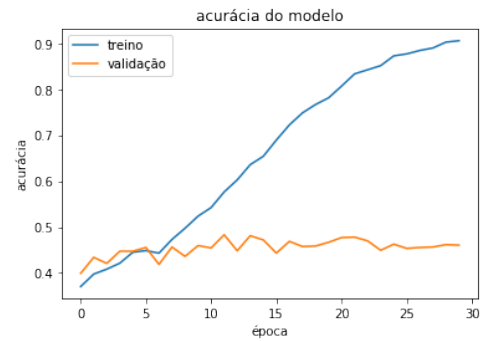


Figura 14. Relação da acurácia através das épocas na CNN inicial com *vertical flip*.

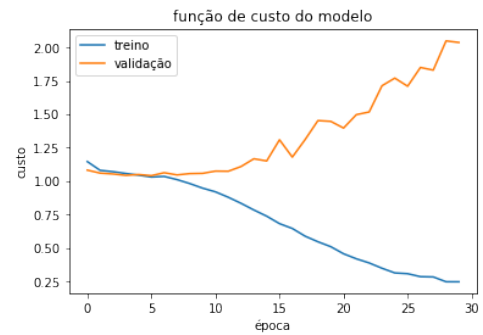


Figura 15. Relação da *loss* através das épocas na CNN inicial usando *vertical flip*.

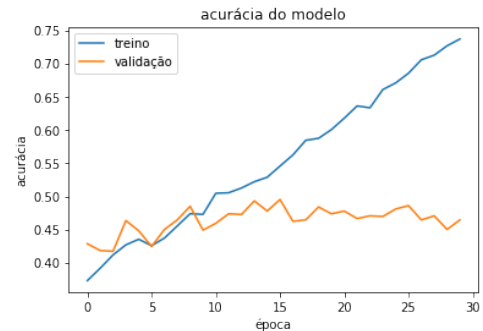


Figura 16. Relação da acurácia através das épocas na CNN usando combinando *horizontal flip* e *vertical flip*.

conjunto de treino e de validação. Na Figura 19 pode-se notar a relação da *loss* com o passar das épocas. Inesperadamente, esse teste teve resultados fortemente insatisfatórios. Entretanto, pela maior diversidade no treino, não observamos o comportamento de *overfitting*.

Esses testes permitiram concluir que a combinação de técnicas de aumentação é interessante para evitar o *overfitting* após poucas épocas de treino. Apesar disso, como muitos testes tiveram resultado abaixo do esperado, acredita-se que a arquitetura da CNN proposta inicialmente não é suficiente para representar o problema em questão. Os melhores resultados obtidos com o uso da VGG16 consolidam essa hipótese.

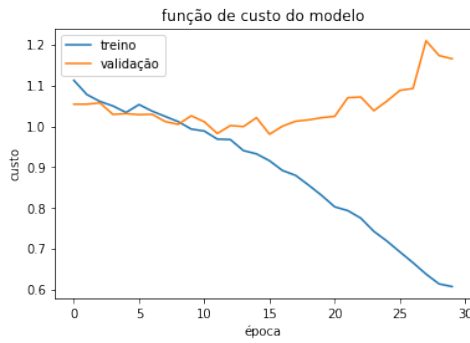


Figura 17. Relação da *loss* através das épocas na CNN inicial combinando *horizontal flip* e *vertical flip*.

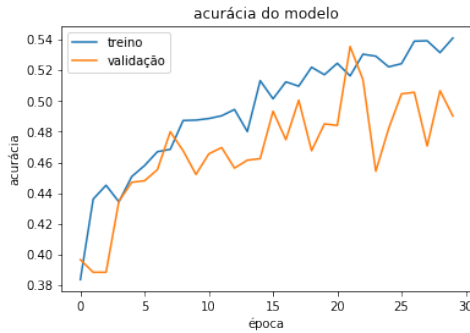


Figura 18. Relação da acurácia através das épocas na rede *Baseline* usando *Data Augmentation* ao rotacionar as imagens do *dataset*

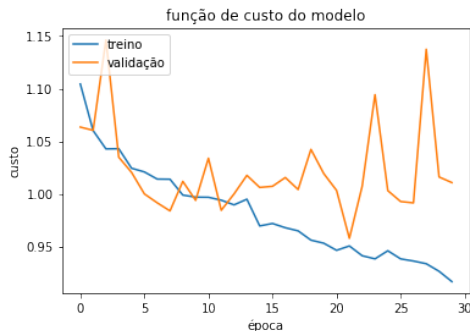


Figura 19. Relação da *loss* através das épocas na rede *Baseline* usando *Data Augmentation* ao rotacionar as imagens do *dataset*.

H. VGG16 com Data Augmentation

Nessa etapa do experimento, combinamos a técnica de *transfer learning* na VGG16 com *data augmentation*. Com a saída proveniente da VGG16, foi aplicada uma camada *flatten*, seguida por uma camada densa de 1024 *neurons* e outra de 512 *neurons*, com função de ativação *ReLU* e *dropout* de 10%, além de última camada de saída *fully connected* com 3 *neurons*, com função de ativação *softmax*. O otimizador utilizado foi o *Adam*, e a rede foi treinada por 30 épocas. Com isso, foi obtida uma acurácia de 46% no conjunto de treino e 33% no conjunto de validação. Podemos conferir a matriz de confusão proveniente do resultado na Figura 20.

	Vegano	Ovo-lacto	Carne
Vegano	14	378	54
Ovo-lacto	5	322	53
Carne	4	341	45

Figura 20. Matriz de confusão utilizando VGG e *data augmentation* com resultados fracos.

	Vegano	Ovo-lacto	Carne
Vegano	222	58	100
Ovo-lacto	93	146	169
Carne	106	103	219

Figura 21. Matriz de confusão utilizando VGG e *data augmentation* com resultados interessantes, acurácia de 50%.

Novamente, a rede constantemente infere a mesma classe para todos os exemplos. Esse comportamento foi especialmente inesperado, pois o uso de *transfer learning* com a VGG16 teve bons resultados, bem como o uso de *Data Augmentation* na CNN inicial. Dessa forma, esperava-se um acurácia superior com a combinação das técnicas.

Uma possível explicação para esse desempenho pode estar na complexidade em se associar padrões distintos, por exemplo, de receitas visualmente distantes, que devem ser associadas a uma mesma classe. Ou seja, além da rede identificar o prato, ela precisa agrupar pratos distintos que sejam vegetarianos estritos, vegetarianos ovo-lactos ou pratos com carne.

IV. CONCLUSÃO E TRABALHOS FUTUROS

Os resultados encontrados para o problema de classificação de alimentações como pertencentes a dietas com restrição de ingredientes se mostrou, conforme esperado, especialmente mais difícil que o problema geral de classificação de imagens, inclusive de pratos. Isso porque a rede precisa identificar o prato e associá-lo à categoria dietética correta.

Apesar de resultados ligeiramente abaixo do esperado, a partir das métricas coletadas para as abordagens experimentadas, foi possível verificar a viabilidade de se utilizar aprendizagem

de máquina para automatizar a classificação de alimentos para dietas restritas.

Não obstante, algumas técnicas se mostraram especialmente promissoras, com resultados razoáveis. É possível que parte da dificuldade em se conseguir acurácia superior para esse problema esteja na necessidade de construir classificadores capazes de identificar os ingredientes e associá-los aos tipos de dieta. Essa tarefa não é trivial, principalmente considerando a anotação dos dados e a necessidade de se deduzir esse tipo de informação exclusivamente a partir da qualidade visual.

Ademais, outro fator que acrescenta a dificuldade ao problema abordado é o fato de diversas receitas vegetarianas mimetizarem a aparência de refeições que tradicionalmente contêm carne. Uma possível solução a ser explorada para esse obstáculo é adicionar outras informações, além da imagética, aos dados.

Portanto, nota-se a necessidade de construção de novos *datasets* mais ricos e diversos para problemas de classificação de alimentos. Isso porque há poucos conjuntos de dados disponíveis com informações nutricionais sobre cada imagem, anotação que é importante e útil para diversos trabalhos, já que em muitos casos, assim como neste experimento, foi preciso aproximar a qualidade de cada ícone a partir do nome da refeição.

REFERÊNCIAS

- [1] T. Dietz, A. S. Frisch, L. Kalof, P. C. Stern, and G. A. Guagnano, "Values and Vegetarianism: An Exploratory Analysis," *Rural Sociology*, vol. 60, pp. 533–542, feb 1995.
- [2] H. C. J. Godfray, P. Aveyard, T. Garnett, J. W. Hall, T. J. Key, J. Lorimer, R. T. Pierrehumbert, P. Scarborough, M. Springmann, and S. A. Jebb, "Meat consumption, health, and the environment," *Science (New York, N.Y.)*, vol. 361, jul 2018.
- [3] A. WORSLEY and G. SKRZYPIEC, "Teenage Vegetarianism: Prevalence, Social and Cognitive Contexts," *Appetite*, vol. 30, pp. 151–170, apr 1998.
- [4] C. T. McEvoy, N. Temple, and J. V. Woodside, "Vegetarian diets, low-meat diets and health: a review," *Public Health Nutrition*, vol. 15, pp. 2287–2294, dec 2012.
- [5] K. Yanai and Y. Kawano, "Food image recognition using deep convolutional network with pre-training and fine-tuning," in *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–6, IEEE, jun 2015.
- [6] U. Tiankaew, P. Chunpongthong, and V. Mettanant, "A Food Photography App with Image Recognition for Thai Food," in *2018 Seventh ICT International Student Project Conference (ICT-ISPC)*, pp. 1–6, IEEE, jul 2018.
- [7] A. Singla, L. Yuan, and T. Ebrahimi, "Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management - MADiMa '16*, (New York, New York, USA), pp. 3–11, ACM Press, 2016.
- [8] F. Zhu, M. Bosch, T. Schap, N. Khanna, D. S. Ebert, C. J. Boushey, and E. J. Delp, "Segmentation assisted food classification for dietary assessment," in *Segmentation assisted food classification for dietary assessment* (C. A. Bouman, I. Pollak, and P. J. Wolfe, eds.), vol. 7873, p. 78730B, International Society for Optics and Photonics, feb 2011.
- [9] F. Ragusa, V. Tomaselli, A. Furnari, S. Battiato, and G. M. Farinella, "Food vs Non-Food Classification," in *Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management - MADiMa '16*, (New York, New York, USA), pp. 77–81, ACM Press, 2016.
- [10] Y. He, C. Xu, N. Khanna, C. J. Boushey, and E. J. Delp, "Analysis of food images: Features and classification," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 2744–2748, IEEE, oct 2014.
- [11] N. Ketkar, "Introduction to Keras," in *Deep Learning with Python*, pp. 97–111, Berkeley, CA: Apress, 2017.
- [12] J. D. Dignam, P. L. Martin, B. S. Shastri, and R. G. Roeder, "TensorFlow: A System for Large-Scale Machine Learning," *Methods in Enzymology*, vol. 101, no. C, pp. 582–598, 2016.
- [13] E. Bisong, "Google Colaboratory," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*, pp. 59–64, Berkeley, CA: Apress, 2019.
- [14] D. Strigl, K. Kofler, and S. Podlipnig, "Performance and Scalability of GPU-Based Convolutional Neural Networks," in *2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, pp. 317–324, IEEE, feb 2010.
- [15] T. Carneiro, R. V. Medeiros Da Nobrega, T. Nepomuceno, G.-B. Bian, V. H. C. De Albuquerque, and P. P. R. Filho, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," *IEEE Access*, vol. 6, pp. 61677–61685, 2018.