

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Relatório Técnico Final

Detecção da Ferrugem em folhas de café usando processamento de
imagens

Uberlândia

31/08/2018

Resumo

Atualmente, existem diversas técnicas para processamento de imagens para construção de classificadores, e seu uso no manejo de precisão dentro da agricultura tem se mostrado bem efetivo, principalmente para a identificação de objetos, como doenças por exemplo, que é uma tarefa difícil para detecção para o olho humano. Essas técnicas podem ser implementadas utilizando técnicas avançadas de *deep learning*, o que os tornam ainda mais eficientes. Nesse projeto foi feita a aquisição das imagens, e estudamos essas técnicas para se implementar um classificador para aplicar ao problema em questão.

1 – Introdução

De acordo com o livro do Syngenta, a ferrugem é uma praga severa para o cafeeiro. A ferrugem é um fungo endofítico que vive em quase todas as folhas de cafeeiros adultos, principalmente nas mais velhas. No Brasil, muitos parques de produção de café são susceptíveis a doença. Segundo Matiello [1], a ferrugem atinge todas as regiões cafeeicultoras do Brasil e se não for controlada adequadamente pode causar uma redução de até 45% na produção do fruto do café.

O Syngenta [2] aponta também sobre o manejo que o produtor deve procurar fazer, e também várias outras considerações a fim de conhecer o vigor da planta e sobre seu potencial produtivo, como também o histórico da doença nos talhões.

Nesse sentido, métodos computacionais podem ser utilizados para avaliar as características presentes nas folhas de café e, conseqüentemente facilitar e agilizar a tarefa de identificação da ferrugem. Existem diversos trabalhos visando detecção de doença em folhas na literatura. No livro de Shaikh [3], cita o desenvolvimento de um robô que pulverizador que utiliza de processamento de imagens para detectar os tipos de doenças para decidir qual pesticida aplicar na folha, por exemplo. Desse modo, este projeto também venha se utilizar de métodos computacionais, porém será processamento de imagens combinado com outra técnica de aprendizado de máquina (do inglês *machine learn*), o *deep learning*.

O *deep learning* ou aprendizado profundo, ou mais comumente chamada de aprendizagem profunda, é um conjunto de algoritmos de aprendizado de máquina que tentam assimilar em vários níveis, correspondendo a diferentes graus de abstração. As técnicas desenvolvidas a partir da aprendizagem profunda tem impactado em diversos trabalhos sobre reconhecimento de padrões envolvendo processamento de imagens, entre outros. Decorrente a esse fato, seu uso em pesquisas vem sido mais constantes [4].

Os níveis desses modelos estatísticos aprendidos correspondem a graus distintos de conceitos, em que os de alto nível são definidos a partir de níveis inferiores, e os mesmos

conceitos de nível inferior podem ajudar a definir muitos conceitos de nível superior. O aprendizado profundo envolve de múltiplos níveis de representação e abstração que ajudam a dar sentido aos dados, como imagens, sons e textos.

No entanto, o aprendizado profundo mostra uma classe muito ampla de técnicas, e arquiteturas de aprendizagem de máquina. Se mostrando também muito eficaz para geração ou classificação, o que fomenta ainda mais o seu uso nesse projeto.

2. Considerações sobre *Deep Learning*

2.1 Redes Neurais

Alguns autores assumem a classificação das técnicas de aprendizado profundo em modelos discriminativos profundos ou em modelos generativos/não-supervisionados. Dentre os discriminativos, são destacadas as Redes Neurais Profundas (do inglês *Deep Neural Network* - DNN), Redes Neurais Recorrentes (do inglês *Recurrent Neural Network* - RNN), Redes Neurais Convolucionais (do inglês *Convolutional Neural Network* - CNN), entre outras estruturas de aprendizado profundo. No entanto, essa maneira de classificação binária deixa escapar um conhecimento oriunda da pesquisa de aprendizagem profunda sobre como modelos generativos ou não supervisionados e podem melhorar bastante a formação de DNNs e outros modelos supervisionada através de melhor regularização ou otimização [5].

A classificação em duas categorias de métodos mostra várias diferenças importantes entre redes profundas de aprendizagem não supervisionada e supervisionada. Comparando tais abordagens, os modelos de aprendizado profundo supervisionados, como as DNNs, são em sua maioria mais eficazes em treinar e testar, e mais flexíveis para construir e se adequam bem para aprendizagem de início a fim de sistemas complexos. Do outro lado, temos que os modelos profundos de aprendizagem não supervisionada, especialmente os de métodos generativos probabilísticos, são mais simples de interpretar, têm menos complexidade para incorporar o problema ser aplicado, são mais diretos em sua composição e para lidar com a incertezas. Porém, os modelos profundos são geralmente intratáveis na interferência e aprendizagem para sistemas complexos [6].

2.2 Classificação com Redes Neurais Convolucionais

Semelhante em sua estrutura com o *perceptron* de múltiplas camadas (MLP), as redes neurais convolucionais (RNC) possuem uma ampla utilização em classificação, reconhecimento e detecção em imagens [7]. Sua estrutura pode se resumir em camadas de convolução, agrupamentos e classificações.

A forma como a estrutura de uma RNC é construída, ou seja, a quantidade de camadas de convolução alternadas com as camadas de agrupamento depende da aplicação.

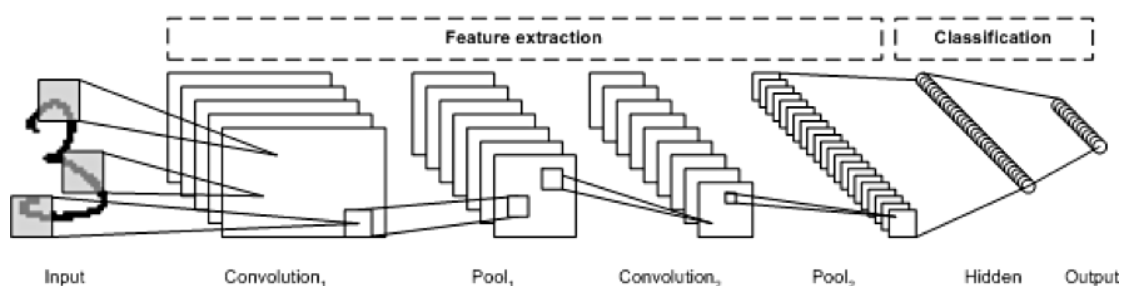


Figura 1 - Exemplo de uma RNC e suas camadas. Fonte: <https://www.ibm.com/developerworks/br/library/cc-machine-learning-deep-learning-architectures/index.html>.

A entrada da RNC é uma imagem de qualquer esquema de cor. A seguir, na camada de convolução cada neurônio está associado a uma janela de *kernel* que será convoluída na imagem de entrada. Esse *kernel* de convolução é composto pelos valores dos pesos de ligação do neurônio. A partir da imagem original são geradas imagens, que resultam das operações em cada um dos n neurônios. Com as novas imagens, devida a operação de convolução, podem ser obtidos valores negativos em alguns pixels. No entanto, é realizada uma operação para a remoção desses valores. As imagens obtidas a partir dessa camada são conhecidas por mapa de características.

Depois da camada de convolução é comum aplicar uma camada de agrupamento (*Pool*). A importância dessa etapa é a redução na dimensionalidade dos mapas de características, acelerando o tempo que é preciso de treinamento da rede.

Na RNC alternam-se camadas de convolução e agrupamento, e após essas camadas que extraem características da imagem original, existe uma camada densa que consiste em uma rede neural convencional para a classificação.

A arquitetura da RNC com essas múltiplas camadas que permite a extração de diversas características da imagem a partir das camadas de convoluções, bordas, círculos, retas, texturas, entre outros, todas elas devido aos valores dos filtros aplicados nessas camadas e que são resultados do *backpropagation* da etapa de treinamento.

2.3 Tensorflow

O *Tensorflow* é uma biblioteca, dentre outras, de código aberto que auxiliam para a área de aprendizado de máquina aplicada a muitos tipos de tarefas. Bastante usado na criação de redes neurais para reconhecimento de padrões e correlações, que são semelhantes à forma humana de aprender.

A biblioteca tem um diferencial pois sua execução possibilita além de serem nas CPUs, poderem ser nas GPUs, que possuem componentes especificamente para cálculos de matrizes tornando-as eficientes para tarefas que incluem processamento de imagens, ou simples matrizes contendo variados tipos de dados. Suas bibliotecas expandem em Python e C++, linguagens de fácil manejo e com bom desempenho, e futuramente chegará em outras como Java, JavaScript, Lua, entre outros.

Desse modo, tendo a sua criação especificamente para a aplicação na área de Inteligência Artificial, que inclui a aprendizagem de máquina e *deep learning*. Além de ter se popularizado na comunidade tem trago bons feedbacks no âmbito da área, e que o destacou para ser escolhido para a aplicação deste projeto.

3. Atividades desenvolvidas

3.1 Aquisição de imagens



Figura 2 - Folha de café doente.

Para a etapa de treinamento da RNC é necessário um banco de dados específico do domínio. No entanto foi criado um banco de dados com as imagens que foram adquiridas manualmente. A figura 2 é um espécime de imagem do banco a ser tratada:

O domínio do projeto em questão, é a detecção da ferrugem na folha de café. Portanto, seriam necessárias uma boa quantidade de imagens de folhas doentes com a doença para quando chegar na etapa de treinamento da arquitetura da RNC ter a base de dados para fazer o treino.

3.2 – Detecção manual da doença

A rede neural é um conjunto de funções nas quais incluem filtros e tratamentos a ser utilizados nas imagens. No entanto, alguns pré-processamentos necessários como a segmentação para fazer uma separação das partes doentes da folha, da folha do restante da imagem, ou seja, um enxugamento da imagem antes de ser entregue a rede neural, principalmente as convolucionais nas quais a entrada é uma imagem. Feita a segmentação é claro o que queremos detectar na folha, o que facilita para classificação da imagem.

3.3 – Preparo das entradas da rede

A partir das imagens do banco feito foi decidido as entradas seriam imagens de entradas $40 \times 40 \times 3$ (40 pixels por 40 pixels com os três canais RGB). Com essa decisão, foram produzidas cerca de 4000 imagens, de diferentes imagens selecionadas do banco que teriam doença e não, para servir de treino e de teste, na qual usaria aproximadamente 3000 para treino e o restante para teste. No conjunto selecionado para treino foi aplicado técnicas de *Data Augmentation*, que consiste em produção de novos dados aplicando rotações, diminuição de tonalidade de cor, tom de luminosidade, entre outros, para aumentar ainda mais o conjunto para que a rede não se acostumassem com os dados de treino. Usando a técnica de aumento de dados a base de treino foi ampliada para cerca de 55000 imagens.

Partindo das imagens de treino e de teste, para cada um foi produzido os seus *labels*, que dizem se a imagem possui ou não doença, que são na verdade as classes de cada imagem, e também um dos parâmetros a ser usados.

3.4 – Modelagem da Rede Neural Convolucional

Para a construção da rede neural convolucional usada neste projeto, foi utilizada de base e adaptada a rede usada para a detecção de reconhecimento de dígitos da MNIST, que é um grande banco de dados de dígitos manuscritos comumente usado para o treinamento de vários sistemas de processamento de imagens como também usado em teste de aprendizado de máquina. A adaptação consistiu na mudança das entradas da rede para se adaptarem as dimensões do conjunto de treinamento criado e seus respectivos *labels*. Na criação da rede, foi feita duas camadas convolucionais ambas com *kernel* 5x5 e a primeira com 32 filtros e a segunda com 64, cada uma seguida por uma camada de *pooling*(agrupamento), finalizando com duas camadas densas a primeira com 2048 neurônios e a segunda com 1024. Para complementar como otimização de resultados da rede, foi observada na estrutura da rede AlexNet que era aplicado uma normalização de dados após a camada de agrupamento. A partir desse ponto, após obter melhores resultados após testes foi colocado implementado na estrutura a técnica de normalização. Além da técnica de aumento de dados, foi aplicada uma outra que o próprio tensorflow implementa, consistida em zerar uma parte dos neurônios para a rede não se acostumar com os dados, usada logo ao fim da ultima camada densa, tendo como uma das entradas a mesma, e escolhida uma taxa de 20% a ser zerada. Por fim a camada de saída, última camada a ter sido construída, que usa como entrada a camada que utiliza a camada densa para aplicar a técnica de zeramento de neurônios, da camada de saída que é mostrado o resultado obtido.

4.Resultados

4.1 Resultados Parciais

Como a entrada da RNC é uma imagem, se espera que a saída também seja uma imagem, no entanto a primeira etapa do projeto foi fazer a marcação manualmente das imagens detectando a localização da doença na folha para ser comparada com a saída da rede e fazer os ajustes dos pesos para otimizar a qualidade de resposta da rede. A seguir os resultados na figura 3 e na figura 4.

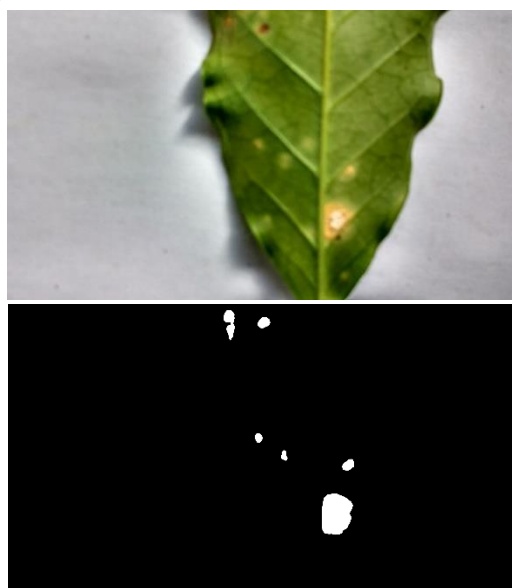


Figura 3 - Exemplo de localização da doença na folha de café por segmentação.

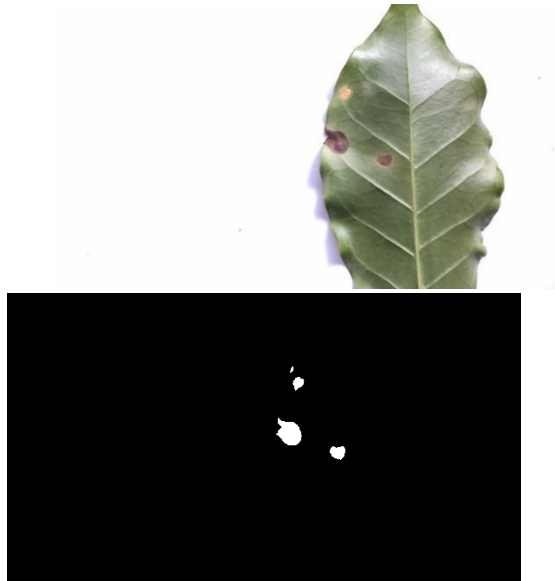


Figura 4 - Exemplo de localização da doença na folha de café por segmentação.

O desenvolvimento da segunda parte do projeto foi possível com o uso do MATLAB que possui diversas funções para processamento de imagens, como funções para obter gráficos, e o poder para lidar com a imagem de diversas formas de acordo com os conhecimentos adquiridos pelo [8]. A ferramenta foi utilizada na tarefa de separar a folha do resto da imagem, pois é a parte interessante para o projeto. Um exemplo na figura 5 do resultado obtido a partir da aplicação na imagem da figura 2:



Figura 5 - Exemplo da separação da folha do restante da imagem.

4.2 – Resultados finais

Para a terceira parte do projeto que foi utilizado das imagens originais, as que foram marcadas manualmente e mascaras binarizadas(em preto branco) mostrando onde era folha. Delas foram tiradas janelas 100x100 que interseccionavam as outras três imagens, na qual deveriam haver janelas com doença, ou somente o corpo da folha, ou alguma delas que poderiam fazendo fronteira com o que não era a parte interessante para o treino, para assim dizer que meu conjunto de treino e teste poderia estar moderadamente distribuído entre as classes com vários modos de aparição. A seguir alguns exemplos das janelas 100x100:



A partir das janelas das imagens originais da do corpo da folha obtidas foram feitas as imagens 40x40 que serviriam de entrada para a rede, tanto para treino como a teste, e que devem ser para ser feito a previsão de qual classe fará parte . A seguir alguns exemplos das imagens:



Figura 7 - Exemplo das entradas obtidas para a rede.

Feito o treinamento com as entradas obtidas que fora a parte mais demoradas depois do próximo resultado, foi feito a etapa de teste para poder avaliar a eficiência da rede, que resultou: na acurácia variando entre 93 94 por cento, com uma taxa de perda que varia entre 10 e 12 por cento. A partir da rede treinada foi feito uma classificação de uma olha inédita inteira, e criado uma nova imagem binarizada mostrando onde foi identificado doença:

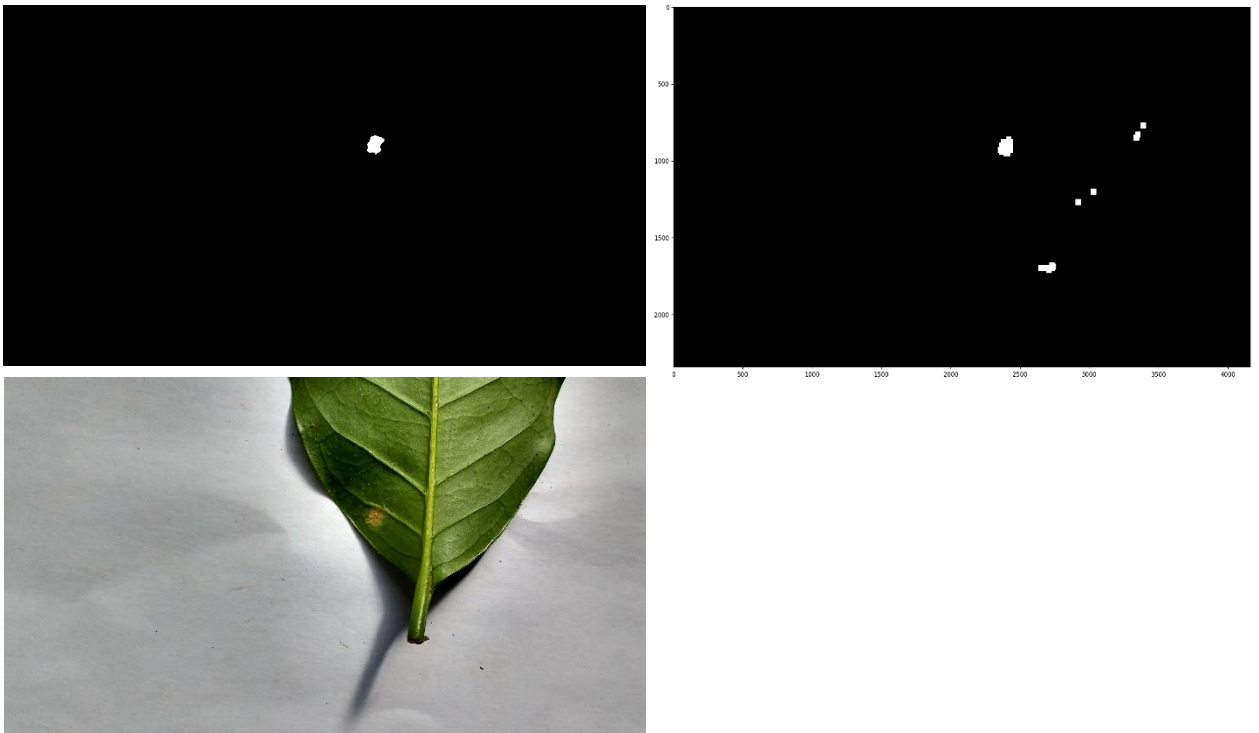


Figura 8 - Exemplo de uma imagem marcada esperada, a imagem que foi produzida a partir do julgamento da rede, e a imagem original de base.

5 – Conclusão

As redes neurais profundas são excelentes classificadores resolvendo de problemas simples a mais complexos implementados por conceitos simples. Para início de conversa é importante visar que ter uma base com boa quantidade de dados, é o primordial e mais importante fato a ser revisado, quando se dirige o problema a usar as RNC para resolvê-lo. Pois quando se recorre para as técnicas de aumento de dados, a base cresce de forma mais pobre se comparado a aumento de imagens de origem único. O processamento de dados é um fato que não se dá para fugir, mesmo quando se desenvolvido uma outra ferramenta (software a parte) para fazê-lo sem ser manualmente, como no caso desse projeto, importante para adaptar os dados a rede. No decorrer do projeto, me deparei com outras ferramentas para o desenvolvimento como o Keras por exemplo, o próprio MATLAB que possuem funções para auxiliar na construção de redes neurais além do Tensorflow. A escolha do Tensorflow foi pela familiaridade no ambiente do python que o projeto seria desenvolvido, juntamente com outras bibliotecas já existentes como o OpenCV(biblioteca para processamento de imagens), e que era de interesse próprio.

6– Referências

- [1] J. B. Matiello, O Café do cultivo ao consumo. Editora Globo S.A., 1st ed., 1991.

- [2] Syngenta, “Dinâmica de pragas e doença do café,” 2012..

- [3] D. A. Shaikh, G. A. G., C. P. A., and K. P. L., “Intelligent autonomous farming robot with plant disease detection using image processing,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 5, no. 4, pp. 1012–1016, 2016

- [4] Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97.

- [5] Nanduri, A. and Sherry, L. (2016). Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). In *Proceedings of 2016 Integrated Communications Navigation and Surveillance (ICNS)*, pages 5C2–1–5C2–8.

- [6] Dean, J., Corrado, G. S., Monga, R., Chen, K., Devin, M., Le, Q. V., Mao, M. Z., Ranzato, M., Senior, A., Tucker, P., Yang, K., and Ng, A. Y. (2012). Large scale distributed deep networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems, NIPS’12*, pages 1223–1231.

- [7] Vargas, A. C. G., Paes, A., and Vasconcelos, C. N. (2016). Um estudo sobre redes neurais convolucionais e sua aplicação em detecção de pedestres. In *Proceedings of*

the XXIX Conference on Graphics, Patterns and Images, pages 1–4. Sociedade Brasileira de Computação.

[8] MARQUES FILHO, Ogê; VIEIRA NETO, Hugo. “Processamento Digital de Imagens”, Rio de Janeiro: Brasport, 1999. ISBN 8574520098.