

# Lab 5 – Representação

Bruno A. Lima

Marcelo G. Pereira

**Professor:** Luiz E. S. Oliveira

*Departamento de Informática*

*Universidade Federal do Paraná - UFPR*

Curitiba, Brasil

**Resumo** – Este trabalho contém a implementação e comparação de métodos para a classificação de textura. Para a extração de características foram utilizados CNN e LBP, e para a classificação o uso de MLP e KNN.

**Index terms** – LBP, CNN, KNN, textura, machine learning, deep learning, rede neural.

## 1. Introdução

O objetivo desse trabalho é a implementação e comparação de métodos para a classificação de 9 tipos de imagens de texturas de plantas, advindas do arquivo macroscopic0.zip da *database* FSD-M [1]. O *dataset* foi dividido em três conjuntos: teste (50%), treino (32,5%) e validação (17,5%).

Diante disso, para a extração de características foram usados *Local Binary Patterns* (LBP) e 2 Redes

Neurais Convolucionais (CNN), sendo elas a VGG16 [2] e outra treinada com o conjunto de treino. Para a classificação, foram utilizados KNN e *MultiLayer Perceptron* (MLP).

## 2. Desenvolvimento

### 2.1 Classificadores

Para a classificação dos vetores de características, foi criada uma rede que é treinada especificamente com os dados de treino resultantes de cada método de extração, a arquitetura da rede pode ser vista em [Fig. 1].

O outro classificador escolhido foi o KNN, o qual teve o parâmetro  $k = 1$  e para o cálculo da distância dos vetores foi usado similaridade de cossenos. Ambas decisões foram com base em testes visando melhorar a acurácia.

A intenção é comparar o desempenho do MLP com o KNN para

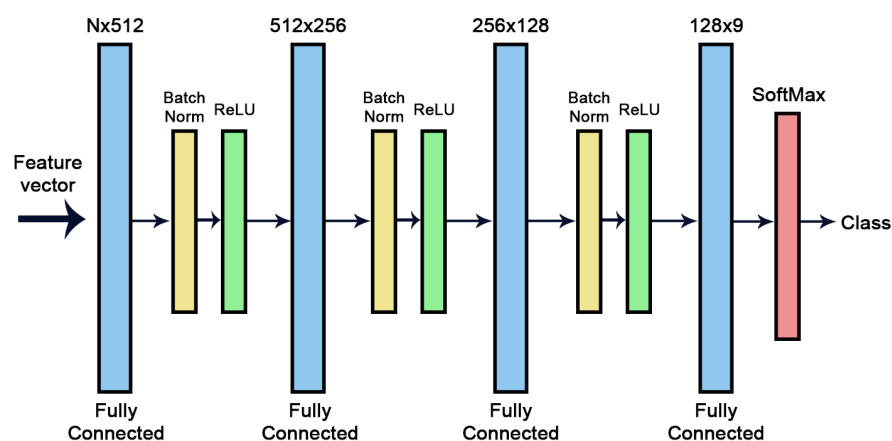


Fig. 1: Arquitetura do MLP usado para classificação

as representações provenientes dos métodos de extração.

## 2.2 Métodos de extração

### 2.2.1 LBP

O método do LBP foi utilizado com 8 vizinhos e raio igual a 3. Após alguns testes, a conversão da imagem para a escala de cinza e o redimensionamento, originalmente 3264x2448, para um tamanho menor, 408x306, mostrou um melhor desempenho, tanto na velocidade do processamento quanto na assertividade posterior nos classificadores.

Além disso, foi gerado um histograma de 63 posições, as quais agrupam os valores resultantes do LBP de 4 em 4. Ao utilizar esse histograma como vetor de características, é notável uma melhora significativa na eficiência. Logo, foi decidido usar o histograma como a representação descendente do LBP.

### 2.2.2 CNN Própria

Para a criação de um modelo próprio, foram treinadas duas CNNs diferentes para extrair características.

As saídas dessas CNNs são concatenadas gerando o vetor de características final [Fig. 2]. O uso de duas CNNs apresentou um desempenho melhor do que apenas uma CNN mais complexa.

Foram utilizadas várias transformações aleatorizadas para aumentar a diversidade dos dados de treino. No total, foram 75 minutos em CPU para completar o treino com 60 *epochs* e *learning rate* de 0.001.

Além disso, as CNNs não extraem características de toda a imagem, apenas de um recorte aleatório (no caso do treino) ou um recorte central (no caso da avaliação) com dimensões 768x768 que é redimensionado para 384x384, melhorando muito o tempo de treinamento e um pouco a acurácia da classificação. Isso vem da ideia de que uma textura tem padrões que se repetem, logo uma amostragem vai conter esse padrão.

Diferente do LBP, o uso de imagens em escala de cinza apresentaram uma perda de acurácia na classificação, por isso são utilizadas imagens RGB.

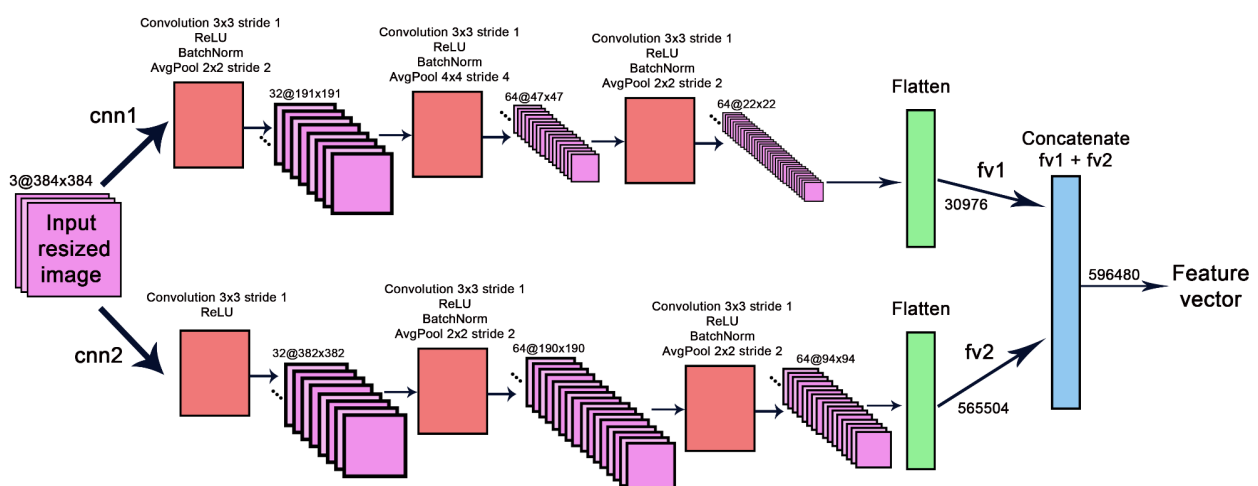


Fig. 2: Arquitetura da CNN própria

### 2.2.3 VGG16

Para a VGG16, foi removido as camadas de classificação, deixando somente a CNN. A partir disso, foram tomadas duas abordagens. A primeira é de congelar a CNN com os pesos pré-treinados e apenas treinar o MLP com as representações decorrentes dela. A segunda é treinar tanto o MLP quanto a parte da CNN.

Os parâmetros para treino foram 60 *epochs* e 0.001 de *learning rate*.

## 3. Resultados

Abaixo está a tabela com a acurácia dos métodos usando cada classificador:

Representação	Acurácia MLP	Acurácia KNN
LBP	91,5%	90,4%
CNN Própria	70,9%	29,7%
VGG16 (frozen)	25,8%	93,2%
VGG16 (fine-tuned)	30,1%	93,2%

A maior acurácia veio da rede VGG16 classificada com KNN, não apresentando diferença das duas abordagens usando esse classificador. No entanto, a abordagem de treinar toda a rede mostrou um melhor desempenho usando o MLP.

O LBP não teve uma grande diferença entre os dois classificadores, tendo um resultado próximo da VGG16 com KNN. Em segundo plano, o LBP apresentou a maior velocidade tanto no treinamento do MLP quanto na classificação com KNN.

A CNN própria não teve um bom resultado com KNN possivelmente pelo vetor de características ser muito grande, o que também deve ter afetado o desempenho do MLP.

## 4. Conclusão

Os resultados alcançados com o treino das redes e do MLP sofrem com a baixa quantidade de imagens destinadas para treino.

Ademais, usar as imagens no tamanho original consome muitos recursos computacionais, ocasionando em um processamento mais lento, além de uma acurácia pior. Dessa forma, utilizar apenas um recorte menor da imagem original providencia uma maior diversidade dos dados e melhora a performance.

Paralelo a isso, uma possível melhoria seria aplicar técnicas para reduzir o tamanho dos vetores de características das CNNs, o que poderia simplificar a classificação e aumentar a acurácia. Visto que para o LBP, o uso de um histograma para a classificação foi mais eficaz do que utilizar diretamente a saída.

## 5. Referências

- [1] L. E. Soares de Oliveira, "Forest Species Database – Macroscopic (FSD-M)", Machine Vision and Applications, vol. 24, nº 4. Zenodo, p. 1019–1031, nov. 23, 2023. Disponível: <https://doi.org/10.5281/zenodo.10219797>
- [2] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014. Disponível: <https://arxiv.org/abs/1409.1556>