



Universidade Federal do Paraná
Bacharelado em Ciência da Computação

Disciplina: Processamento de Imagens [CI1394]

Relatório referente ao
Laboratório 5: Representação

João Pedro Vicente Ramalho - GRR20224169
Luan Marko Kujavski - GR20221236

Maio - 2024

1. Introdução

Neste relatório, abordaremos as etapas realizadas ao longo do experimento, juntamente com as definições dos tópicos relevantes para a compreensão e sua reprodutibilidade. O objetivo deste experimento é realizar a classificação de imagens de textura utilizando algumas técnicas de representação de imagens e algoritmos de aprendizado de máquina. Para isso, utilizaremos a base de dados FSD-M, explorando diferentes métodos, como Local Binary Patterns (LBP) e Redes Neurais Convolucionais (CNN), e avaliando o desempenho dessas abordagens por meio de métricas de avaliação padronizadas.

Além disso, exploraremos os algoritmos de classificação utilizados e a metodologia empregada para avaliar o desempenho dessas abordagens.

2. Tópicos Auxiliares

Esta seção é destinada a uma breve descrição do conjunto de tópicos utilizados para o desenvolvimento das etapas relacionadas ao treinamento e otimização dos modelos de classificação de imagens de textura.

2.1. Overfitting

O *Overfitting* consiste em um problema comum em aprendizado de máquina e modelagem estatística, no qual um modelo se ajusta muito bem aos dados de treinamento, porém falha em generalizar para novos dados, como os dados de teste ou de produção. Isso significa que o modelo aprendeu não apenas os padrões subjacentes nos dados, mas também o ruído e as peculiaridades específicas do conjunto de treinamento.

Para garantir que o modelo mantivesse um bom desempenho em dados novos, foram adotadas várias medidas de regularização. Uma dessas técnicas foi o *dropout*, que consiste em desativar aleatoriamente uma proporção de neurônios durante o treinamento, forçando o modelo a não depender excessivamente de características específicas e promovendo uma maior robustez.

2.2. Learning Rate

A taxa de aprendizado, ou Learning Rate, é um dos parâmetros configuráveis do treinamento de redes neurais que impacta tanto a convergência quanto o desempenho do modelo. Selecionar e ajustar adequadamente essa taxa é essencial para um treinamento eficiente, pois ela determina a velocidade e a precisão com que o modelo aprende a partir dos dados.

2.3. Data Augmentation

Data Augmentation é uma técnica usada em aprendizado de máquina e processamento de imagens para aumentar a quantidade de dados disponíveis para treinamento, gerando novas amostras de treinamento a partir das amostras originais por meio de transformações aleatórias e controladas. Essas transformações incluem rotações, translações, inversões horizontais ou verticais, mudanças de brilho, etc. O objetivo principal do *data augmentation* é melhorar a generalização e a robustez do modelo ao expor o algoritmo a uma variedade maior de variações nos dados de treinamento.

2.4. Local Binary Patterns (LBP)

Local Binary Patterns (LBP) é uma técnica de descrição de textura em imagens amplamente utilizada em visão computacional e reconhecimento de padrões. A premissa por trás do LBP é extrair características discriminativas de uma imagem examinando as relações locais entre os pixels. Em vez de considerar os valores absolutos dos pixels, o LBP examina as diferenças de intensidade entre um pixel central e seus vizinhos imediatos. Tal processo é ilustrado na Figura 1.

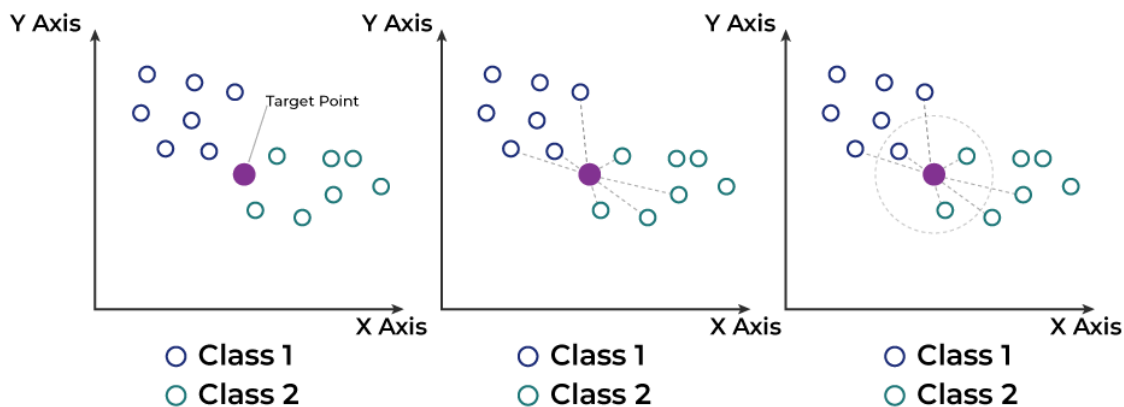


Figura 2: Esquematização do k -NN. Retirado de [2].

3. Configuração da Base de Dados

O conjunto de imagens adotado é composto por 557 imagens, distribuídas em 9 classes. O particionamento dos dados foi implementado respeitando a seguinte regra: 50% para treino, 30% para teste e os 20% restantes foram destinados à validação.

Nessa etapa de configuração optamos por aplicar a técnica de *data augmentation* de modo a segmentar cada imagem em nove partes. Testamos, assim, modelos treinados com um *dataset* de imagens inteiras e o outro com imagens particionadas.

4. Resultados

Esta seção introduz os classificadores empregados neste trabalho, fornecendo uma breve descrição das etapas adotadas para garantir a replicabilidade dos experimentos. Visando aprofundar nosso estudo, adotamos duas abordagens distintas e comparamos seus resultados.

A primeira técnica consiste na utilização de um conjunto de imagens, sem aplicação de zoneamento, que posteriormente será submetido à técnica de particionamento abordada na Seção 3. Nestas condições, as imagens são redimensionadas para o tamanho 224x224 e submetidas à técnica de representação que está sob análise (LBP, VGG ou ResNet), por meio da qual obtemos os vetores de características. A

classificação das imagens foi conduzida utilizando o algoritmo k -NN com distância euclidiana, empregando $k=1$.

A segunda técnica segue uma metodologia semelhante à primeira, com a única diferença sendo a etapa de pré-processamento das imagens descrita na Figura 3.

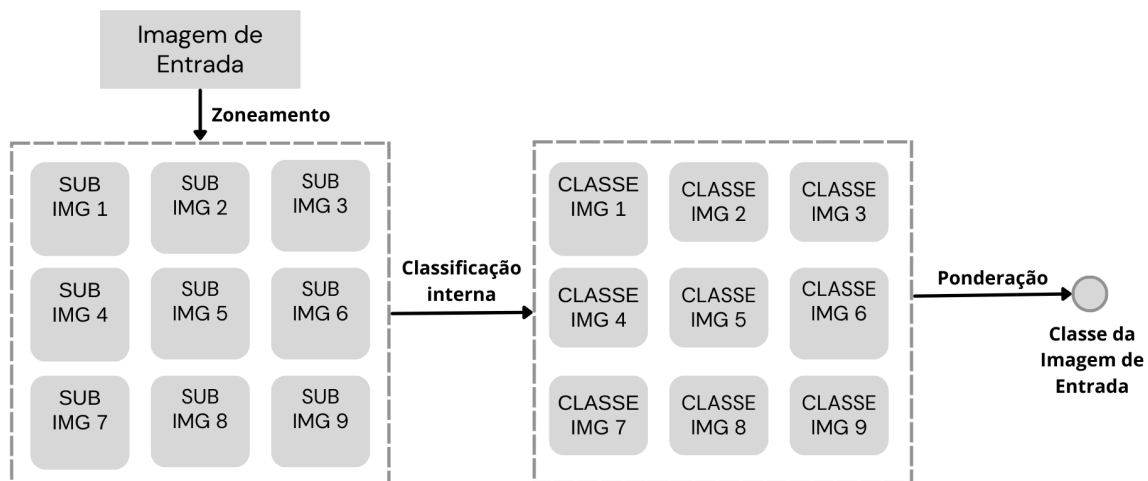


Figura 3: Metodologia de zoneamento para classificação.

Como representado pela Figura 3, a imagem de entrada é zoneada em nove sub-regiões de tamanhos iguais e cada uma é submetida ao algoritmo classificador k -NN. A categorização da imagem original ocorre por meio da regra da maioria, ou seja, a classificação final é determinada com base na maioria das classificações atribuídas às sub-regiões.

4.1. Classificação por LBP

Para categorizar um conjunto de imagens utilizando o LBP, consideramos um raio de um pixel e o número de pontos vizinhos igual a oito. Utilizamos a função *local_binary_pattern* da biblioteca Skimage com o método “nri_uniform”, que retorna, para cada imagem em escala de cinza, um vetor de características com 59 posições. Em seguida, geramos um histograma para cada imagem.

Com todos os histogramas gerados, alimentamos um algoritmo k -NN com os dados de treino e, a partir disso, classificamos cada dado de teste utilizando a distância euclidiana.

A aplicação do primeiro método proposto resultou em uma acurácia de 67,46%. Contudo, com o pré-processamento de zoneamento a acurácia aumentou para 72,28%.

A comparação entre as abordagens demonstra que a segmentação das imagens foi responsável pela melhora significativa dos acertos, evidenciando a importância do aumento da quantidade de dados de treino na performance do algoritmo, pois, após a segmentação das imagens o número de dados de treino aumentou em nove vezes.

4.2. Classificação por VGG pré-treinada

A cargo de contextualização, VGG16 é uma rede neural convolucional (CNN) desenvolvida pelo Visual Geometry Group (VGG) da Universidade de Oxford, e foi apresentada pela primeira vez em 2014. Esta arquitetura é notável pela sua simplicidade e eficácia em tarefas de reconhecimento de imagem.

Para a implementação dessa rede neural utilizamos a biblioteca Tensorflow e replicamos o estudo da técnica de classificação com e sem a segmentação das imagens, mencionadas na Seção 4, todas com modelos pré-treinados no dataset ImageNet.

Diferentemente do LBP, a aplicação do VGG para obtenção dos vetores de características gerou melhores resultados para a primeira abordagem, que apresentou acurácia de 89,15%, enquanto a segunda obteve 77,71% de acerto na classificação.

Acreditamos que a diminuição da acurácia na segunda abordagem, apesar da aplicação do zoneamento, pode ser atribuída à inconsistência entre as partes segmentadas das imagens. Ao dividir uma imagem em nove partes e classificá-las individualmente, o modelo pode encontrar dificuldades para reconciliar as diferentes características e contextos presentes em cada parte. Isso pode levar a uma votação majoritária menos confiável na determinação da classe da imagem original, resultando em uma diminuição da acurácia geral do modelo. Além disso, a segmentação pode levar à perda de informações globais

importantes, prejudicando ainda mais a capacidade do modelo de realizar uma classificação precisa.

4.3. Classificação por ResNet50

A ResNet50 é uma rede neural convolucional profunda desenvolvida pelos pesquisadores da Microsoft Research e apresentada em 2015. Foram examinados quatro métodos durante a avaliação:

1. Modelo pré-treinado usando imagens inteiras
2. Modelo pré-treinado usando imagens segmentadas
3. Treinamento do modelo usando imagens inteiras
4. Treinamento do modelo usando imagens segmentadas

4.3.1. Modelo pré-treinado usando imagens inteiras

Para a primeira abordagem, executamos um modelo pré-treinado no dataset ImageNet em dados de treinos e obtivemos uma acurácia de 97,14%.

4.3.2. Modelo pré-treinado usando imagens segmentadas

Para a segunda abordagem, também utilizamos um modelo pré-treinado no dataset ImageNet. Como saída obtivemos 97,59% de acerto.

4.3.3. Treinamento do modelo usando imagens inteiras

Para treinarmos um modelo, fizemos ajustes à camada densa de um modelo pré-treinado no dataset ImageNet. Assim, testamos cinco parâmetros para esta técnica. Em todos os casos, as imagens foram redimensionadas para 224x224 para o treino e utilizamos 100 epochs no treinamento. Os resultados estão representados pela Tabela 1.

Modelo	Dropout	Acurácia
--------	---------	----------

1	0	97,59%
2	0.1	98,19%
3	0.2	96,98%
4	0.3	93,97%
5	0.4	95,18%

Tabela 1: Resultados do treinamento sem segmentação.

Como o segundo modelo foi o melhor no dataset de teste, exploramos adicionar *data augmentation* aos seus dados de treinamento. Adicionamos *RandomFlip* horizontal e vertical, *RandomZoom* de 20% e *RandomContrast* de 20%. Porém, a acurácia diminuiu para 96,98%.

É importante mencionar que devido à quantidade reduzida de imagens disponíveis, treinamos somente as camadas densas da rede neural. Apesar desse ajuste, a acurácia diminuiu em razão de sua generalização.

4.3.4. Treinamento do modelo usando imagens segmentadas

Para esse caso, também utilizamos um modelo pré-treinado no dataset ImageNet e ajustamos a camada densa ao nosso dataset. Em todos os casos, as imagens foram redimensionadas para 224x224 para o treino e utilizamos 100 epochs no treinamento. Também testamos aplicar técnicas de *data augmentation*, como *RandomFlip* horizontal e vertical, *RandomZoom* de 20% e *RandomContrast* de 20%.

Para testarmos, utilizamos a regra de soma de probabilidades posteriores, assim a classe resultando é a que apresenta maior soma de probabilidades posteriores extraídas do modelo treinado. Tal aplicação é exemplificada pela Figura 4.

Sub-imagem	Classes			Classificação
	1	2	3	
1	0,60	0,20	0,20	1
2	0,20	0,30	0,50	3
3	0,30	0,30	0,40	3
4	0,50	0,20	0,30	1
5	0,40	0,15	0,45	3
6	0,10	0,30	0,60	3
7	0,10	0,50	0,40	2
8	0,20	0,55	0,25	2
9	0,30	0,10	0,60	3
Soma	2,70	2,60	3,70	3

Figura 4: Exemplo da soma de probabilidades posteriores. Adaptado de [3]

Como podemos observar na Figura 4, a imagem de entrada foi segmentada em 9 sub-imagens. Cada sub-imagem foi submetida à classificação e obtivemos três valores correspondentes à possibilidade dela pertencer a Classe 1, Classe 2 e Classe 3, respectivamente. Por fim, após somar as probabilidades concluímos que a imagem original pertence à Classe 3, pois sua soma sobressai às demais.

Para o nosso treinamento do modelo usando imagens segmentadas, obtivemos os resultados da Tabela 2.

***Data Augmentation: RandomFlip, RandomZoom e RandomContrast**

Modelo	Dropout	Data Augmentation*	Acurácia
1	0	Não	98.19%
2	0	Sim	97,59%
3	0.1	Não	98,19%
4	0.1	Sim	98,79%
5	0.2	Não	98,79%
6	0.2	Sim	98,79%

Tabela 2: Resultados do treinamento sem segmentação.

Os resultados obtidos desse treinamento indicam que a aplicação de técnicas de data augmentation e a variação do dropout têm um impacto significativo na acurácia do modelo. Por outro lado, modelos sem qualquer forma de data augmentation geralmente apresentam desempenhos inferiores.

Isso sugere que, enquanto o aumento de dados contribui para a robustez do modelo, uma configuração equilibrada de dropout também é crucial para evitar o overfitting e melhorar a generalização. Por fim, concluímos que a combinação adequada de técnicas de aumento de dados e a otimização dos parâmetros de dropout são essenciais para maximizar a precisão do modelo ao lidar com imagens segmentadas

5. Conclusão

Neste relatório, exploramos uma série de técnicas de representação de imagens e algoritmos de classificação para a tarefa de reconhecimento de texturas, utilizando a base de dados FSD-M. Ao comparar abordagens com e sem segmentação de imagens, observamos que a segmentação pode influenciar significativamente a acurácia do modelo, dependendo da técnica utilizada. Para o LBP, a segmentação das imagens em nove partes aumentou a acurácia ao ampliar a quantidade de dados de treino.

No caso das redes neurais convolucionais, como VGG16 e ResNet50, a segmentação teve diferentes efeitos. Enquanto a VGG16 apresentou uma diminuição na acurácia devido à inconsistência entre as partes segmentadas, a ResNet50 mostrou uma melhoria considerável, alcançando até 98,79% de acurácia quando combinada com técnicas de data augmentation e uma boa configuração de dropout.

Estes resultados ressaltam a importância da escolha cuidadosa dos parâmetros configuráveis do treinamento de redes neurais e das técnicas de pré-processamento, evidenciando que a metodologia deve ser adequada à natureza dos dados e à arquitetura do modelo para maximizar a performance em tarefas de classificação de texturas.

6. Referências

- [1] He, Q., Li, C., & Zhang, Y. Visualization of calculation of Local Binary Pattern (LBP) [Diagrama explicativo do cálculo do LBP].
- [2] GeeksforGeeks. K-Nearest Neighbours [Ilustração do algoritmo k-nearest neighbours].
- [3] Paula Filho, P. L., Oliveira, L. S., Nisgoski, S., Britto Jr., A. S. Forest species recognition using macroscopic images. Original Paper. Springer-Verlag Berlin Heidelberg, 2014.