



Validação de Assinaturas Utilizando Redes Neurais e Árvore de Decisão

Reconhecimento de Padrões

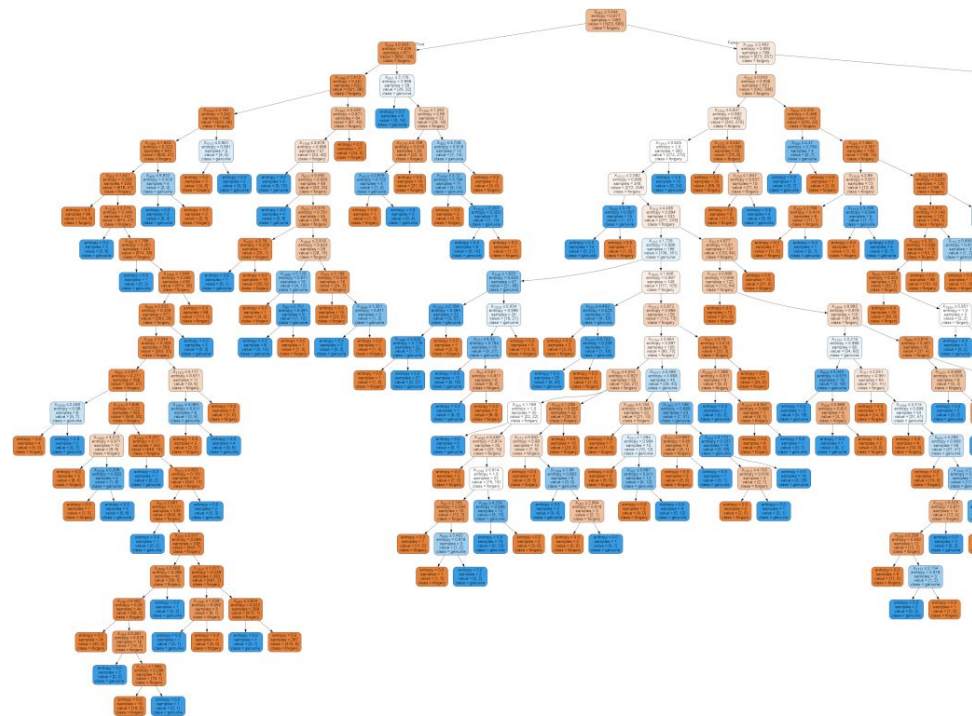
Arthur Flor e Jessiane Pereira
{afsn, jsmp}@ecomp.poli.br

Conteúdo

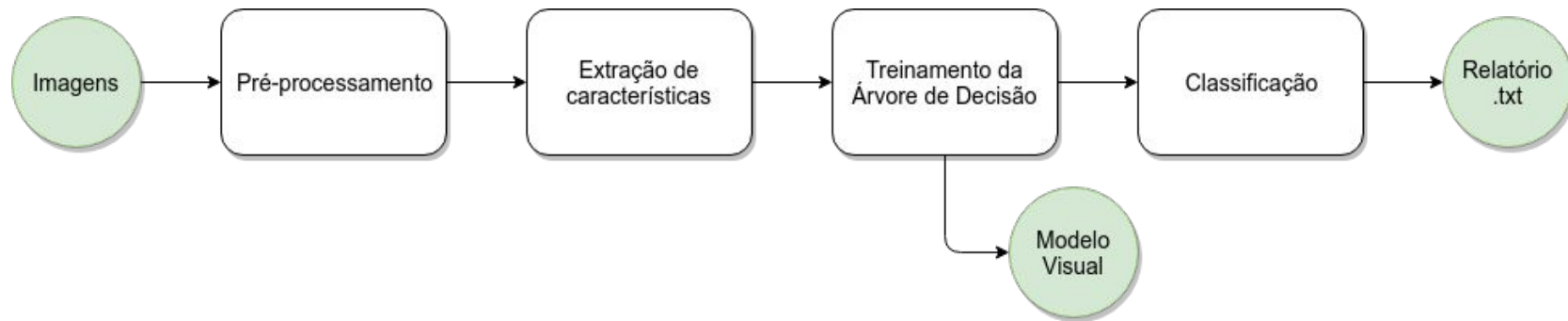
1. Introdução
2. Fluxograma do Projeto
3. Elementos do Fluxograma
4. Ferramentas
5. Resultados
6. Conclusões
7. Referências

Introdução

No contexto do estudo de reconhecimento de escrita, o projeto utiliza a técnica de Árvore de Decisão para classificar imagens de assinaturas, buscando identificar se a escrita é autêntica ou forjada.



Fluxograma do Projeto



Dados Utilizados

- SigComp2009
- Última atualização em 2012
- 3462 imagens de assinaturas (tamanho variado)
- 2 classes (autêntica, forjada)



Exemplo das imagens do dataset

Pré-processamento

1. Binarização da imagem
2. Minimum Bounding Box no caractere
3. Redimensionamento da imagem para 512x512

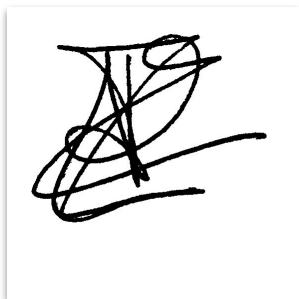
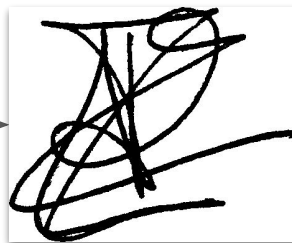


Imagem 587x583



*Binarização e Minimum
Bounding Box*

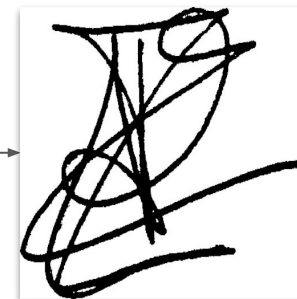
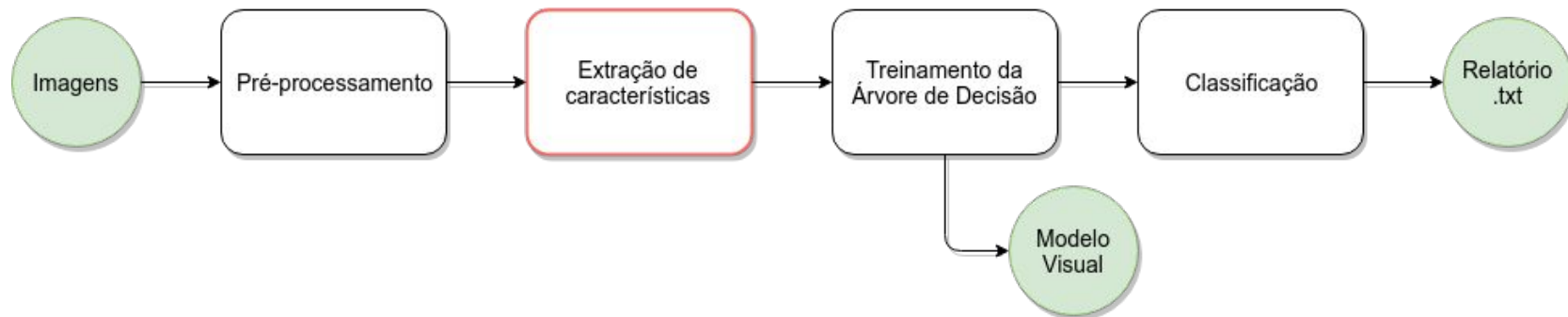


Imagem 512x512

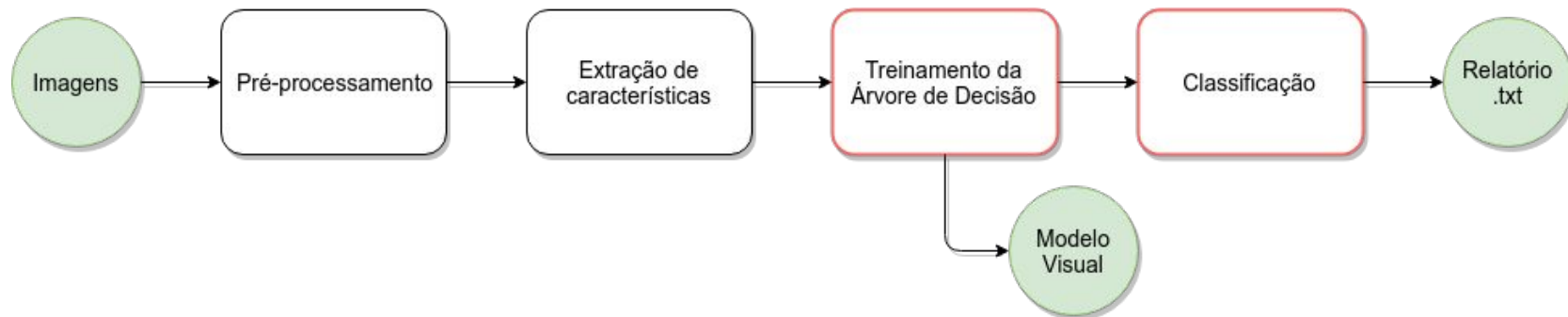
Extração de Características

- Momentos Invariantes de Hu (vetor com 7 características)
- Convolutional Neural Network - CNN (vetor com 2048 características)



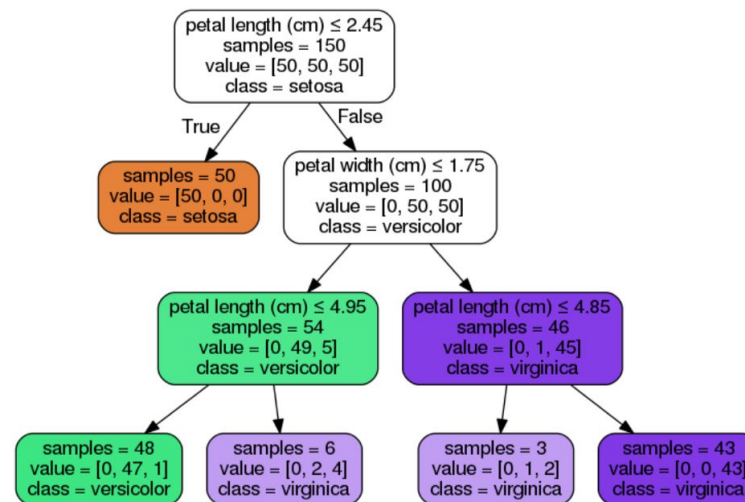
Algoritmos para Treinamento/Classificação

- Implementação manual: C4.5
- Implementação com sklearn: CART e Random Forest



Ferramentas

- Python
- Scikit-learn
- Theano/Lasagne
- Graphviz



Exemplo do modelo visual gerado pela biblioteca Graphviz

Experimento

- Executar todas as combinações das técnicas de extração de características e de árvore de decisão (6 combinações);
- Dividir as imagens, de modo aleatório, em 66,67% para treinamento e 33,33% para teste;
- Realizar 30 vezes rodadas para cada combinação, armazenando o tempo de execução e a taxa de acerto.

Resultados (Precisão)

Treinamento: 2308 imagens

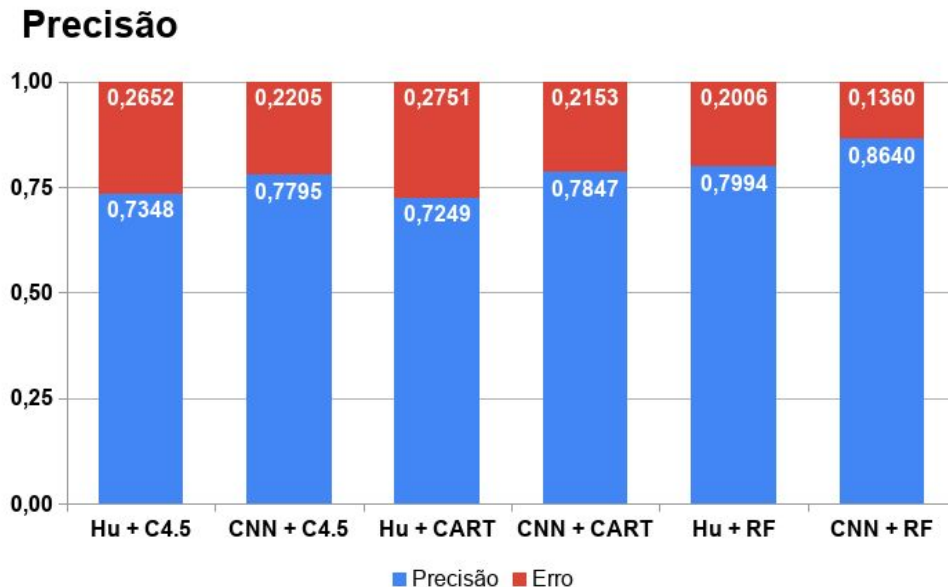
Teste: 1154 imagens

Máx:

CNN + RF (86,40%): 997/1154

Mín:

Hu + CART (72,49%): 837/1154



Resultados (Tempo)

Treinamento: 2308 imagens

Teste: 1154 imagens

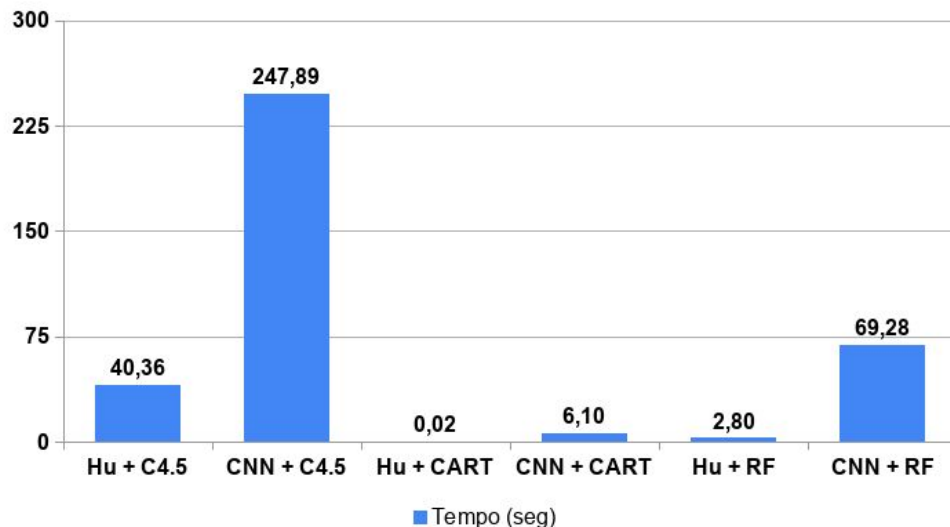
Máx:

CNN + C4.5: 247,89s

Mín:

Hu + CART: 0,02s

Tempo de execução



Conclusões

- A taxa de acerto melhora conforme o pré-processamento;
- A árvore de decisão obteve um melhor resultado quando combinada com a CNN;
- O código implementado manualmente (C4.5) não está otimizado. Isso foi observado no tempo de execução quando combinado a CNN (input de 2048 características).

Referências

- [http://www.iapr-tc11.org/mediawiki/index.php?title=ICDAR_2009_Signature_Verification_Competition_\(SigComp2009\)](http://www.iapr-tc11.org/mediawiki/index.php?title=ICDAR_2009_Signature_Verification_Competition_(SigComp2009))
- <https://machinelearningmastery.com/implement-decision-tree-algorithm-scratch-python/>
- <http://scikit-learn.org/stable/modules/tree.html>
- <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>
- https://github.com/luizgh/sigver_wiwd
- <https://medium.com/@williamkoehrsen/random-forest-simple-explanation-377895a60d2d>
- <https://towardsdatascience.com/random-forest-in-python-24d0893d51c0>
- <https://www.graphviz.org/>